Tastiera

manuale d'uso

INFORMAZIONI IMPORTANTI

Il Sistema di gioco ATARI XE utilizza e produce energia a radio frequenza. Se non viene installato ed usato seguendo le istruzioni riportate in questo manuale, l'apparecchio può causare interferenze alla ricezione dei vostri apparecchi radiotelevisivi.

Se rilevate la presenza di interferenze mentre usate l'apparecchio, spegnetelo. Se le interferenze cessano, è probabilmente l'apparecchio a provocarle. Con l'apparecchio acceso, cercate di risolvere il problema applicando le misure che seguono:

* Modificare la posizione dell'antenna del ricevitore.
* Spostare l'apparecchio in funzione del ricevitore.
* Allontanare l'apparecchio dal ricevitore.
* Collegare l'apparecchio in una diversa presa di corrente elettrica in modo che sia collegato su un ramo diverso del circuito rispetto al ricevitore.

Consultate il vostro negoziante ATARI o un esperto riparatore radio-TV per ulteriori consigli.

ATTENZIONE Questo apparecchio è conforme alle norme riguartdanti gli elaboratori elettronici di classe B e alla sezione 15 paragrafo J delle nòrme FCC. Questi regolamenti prevedono una ragionevole protezione contro le interferenze nel caso l'apparecchio venga usato in ambienti residenziali. Comunque, non c'è alcuna garanzia che non si verifichino interferenze in perticolari situazioni. Solo le periferiche (unità d'ingresso/uscita, terminali, stampanti, ecc.) conformi ai limiti suddetti possono essere collegate a questo computer. L'uso di periferiche non conformi probabilmente genererà interferenze con la ricezione dei segnali radio e TV. E' necessario usare cavi schermati su tutti i connettori di ingresso/uscita (eccetto i connettori joystick e mouse), altrimenti le emissioni di radiofrequenza potrebbero eccedere i limiti delle norme FCC.

E' stato fatto ogni sforzo possibile per assicurare l'accuratezza della documentazione del prodotto descritto in questo manuale. Comunque, poichè la Atari Corporation migliora ed aggiorna costantemente il software e hardware dei suoi computers, è impossibile garantire l'accuratezza del manuale a partire dalla data di pubblicazione, e quindi sono possibili cambiamenti, e che si verifichino errori di stampa o omissioni. La riproduzione di tutto o di parte di questo manuale è vietata senza preventiva autorizzazione scritta della Atari Corporation.

Stampato in Italia nel marzo 1988 dalla Black-out Editrice - Modena

Tastiera

manuale d'uso

INDICE

INTRODUZIONE

Approccio con la tastiera ATARI XE

La tastiera ATARI XE trasforma il vostro XE in un sistema molto sofisticato.

Con la tastiera collegata alla console, potete usare per creare i vostri programmi il BASIC ATARI oppure il linguaggio di programmazione residente XE. E, se siete un giocatore serio, aggiungendo la tastiera, potrete utilizzare i più sofisticati giochi per computer che richiedono l'intererazione della tastiera.

La possibilità di collegare sia un registratore a cassette che un disk drive, vi permette di utilizzare giochi e programmi applicativi disponibili sia su disco che su cassetta. Ora avete il vantaggio di poter utilizzare elaboratori di testi, programmi di archiviazione dati e molti altri programmi - tutto con il vostro sistema XE.

Utilizzo del manuale

E' semplice operare con la tastiera XE e utilizzare il BASIC ATARI, e avrete i migliori risultati se conoscete la tastiera e il BASIC correttamente fin dall'inizio. Questo manuale spiega come fare questo in modo chiaro, con un linguaggio non tecnico che i novizi del computer possono capire.

I capitoli in questo manuale sono stati strutturati in modo tale da essere letti in sequenza. Cominciate collegando la tastiera e così via fino ad imparare le caratteristiche della tastiera. I rimanenti capitoli offrono una guida sull'ATARI BASIC per i programmatori BASIC che iniziano per la prima volta.

Capitolo 1: Per iniziare spiega come collegare la tastiera XE alla console e attivare il sistema. Imparerete anche ad utilizzare la tastiera per visualizzare caratteri grafici speciali e caratteri di lingue internazionali.

Capitolo 2: Editare programmi BASIC con la tastiera ATARI XE spiega come utilizzare la tastiera per inserire ed editare programmi BASIC ATARI.

Capitolo 3: Scrivere un semplice programma BASIC dimostra passo dopo passo come creare il vostro primo programma BASIC.

Capitolo 4: Creare un Loop Interattivo in BASIC che vi spiega come utilizzare i comandi BASIC in modo automatico per la ripetizione automatica di segmenti di programmi.

Capitolo 5: Elaborazioni di numeri casuali e funzioni matematiche col BASIC vi spiega come fare calcoli matematici all'interno di programmi BASIC.

Capitolo 6: Realizzare decisioni e risolvere problemi col BASIC vi spiega come scrivere programmi che simulino il ragionamento umano di approccio alle decisioni nella risoluzione di un problema.

Capitolo 7: Produzione di suoni e grafici col BASIC vi spiega come far suonare i vostri programmi con anche dei grafici molto colorati, musica, e altri suoni.

Appendice A: Semplici programmi BASIC vi offre un campionario di programmi BASIC da provare.

Appendice B: Parole riservate BASIC è una lista di comandi BASIC e relativa descrizione delle loro funzioni.

Appendice C: Set di Caratteri ATASCII è una lista di tutti i caratteri disponibili con la tastiera XE.

Appendice D: Messaggi di Errore è una lista di codici di errori BASIC e il loro significato.

L' Indice vi aiuta a localizzare i termini e le procedure spiegate nel manuale.

Supporto per l'Utente vi dice dove trovare ulteriori informazioni sui prodotti Atari.

I paragrafi contrassegnati **Nota** o **Attenzione** vengono visualizzati di volta in volta nel manuale. Le note contengono utili informazioni riguardo ai temi discussi nel capitolo. Attenzione vi segnala potenziali problemi e vi suggerisce i modi di risolverli.

Nota: in questo manuale, i caratteri racchiusi in parentesi quadre ([]) rappresentano tasti. Alcune volte una procedura vi richiede di utilizzare due tasti contemporaneamente. In questo caso, i tasti sono elencati in ordine. Per esempio **[Opzione][Select]** significa di premere e tenere premuto il tasto **[Opzione]** e premere il tasto **[Select]** per effettuare la funzione.

CAPITOLO 1
PER INIZIARE

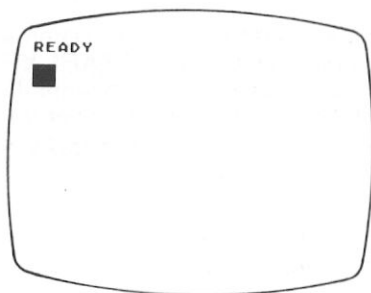
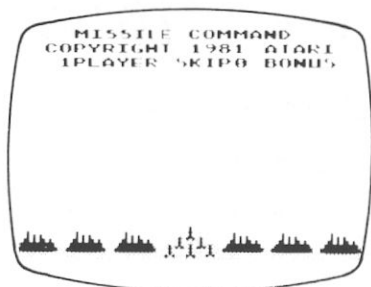
Collegare la tastiera XE

Con la tastiera collegata al vostro sistema di giochi XE, potete utilizzare il linguaggio di programmazione residente XE, ATARI BASIC, e allo stesso tempo il gioco Missile Command e altri giochi che utilizzano il controllo della tastiera.

Prima di collegare la tastiera XE, inizializzate il vostro sistema di giochi XE come descritto nel manuale d'utente **ATARI XE Game System**. Assicuratevi che la console XE sia spenta (la luce dell'indicatore sul tasto **[Power]** deve essere spenta). Quindi collegate il cavo della tastiera XE alla porta della tastiera sul lato sinistro della console.

Il retro della tastiera ha ampie prese per il collegamento alla console. Per collegare le due unità, allacciate il fronte della tastiera con la console attraverso i due ganci di plastica.

Accendere il vostro sistema di giochi XE e la tastiera

Una volta che avete collegato una tastiera, potete usare uno dei due programmi residenti XE, ATARI BASIC o Missile Command. Per lanciare il BASIC, accendete semplicemente la vostra console (premete il tasto **[Power]**). Con la tastiera collegata, XE carica il BASIC al posto del Missile Command e lo visualizza sullo schermo.

Per caricare il Missile Command, tenere premuto il tasto **[Select]** quindi accendete la console. Il titolo Missile Command apparirà sullo schermo.

Azionare il sistema con un Disk Drive

Collegando un Disk Drive Atari al XE vi permette di utilizzare i giochi basati su disco e le loro applicazioni. Potete inoltre inizializzare il vostro sistema con un programma su disco al posto del Missile Command o del BASIC.

Per inizializzare il vostro sistema con un programma su disco quando un Disk Drive è collegato seguite i passaggi qui riportati:

1. Accendete il vostro video e il vostro Disk Drive
2. Quando la luce del drive busy si spegne, inserite il disco programma che state per utilizzare nel drive e chiudete la porta del drive
3. Accendete l'alimentazione della console. Se il vostro programma viene lanciato col BASIC, semplicemente premte **[Power]**. Se il vostro programma viene lanciato senza BASIC, tenete premuto **[Opzion]** e premete **[Power]**.

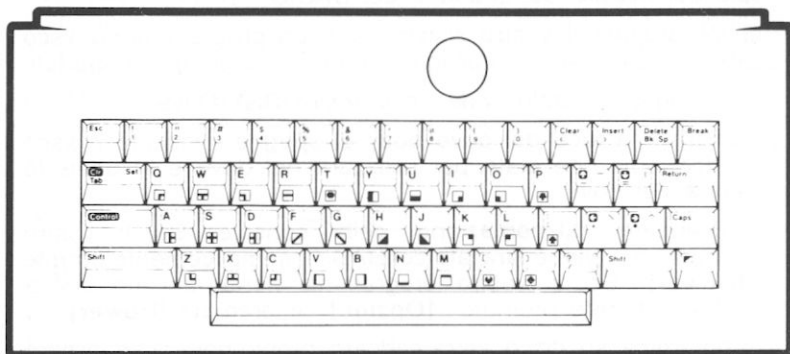
Il programma su disco verrà caricato, pronto per voi ad iniziare il lavoro.

Lanciare programmi su cassetta

Collegando un registratore Atari al XE vi permette di utilizzare programmi forniti su cassette. Per lanciare un programma su cassetta, seguite questi passaggi:

1. Collegate un registratore al vostro sistema XE allacciando il suo cavo nella porta SIO sul retro della console. Assicuratevi che la vostra tastiera sia anch'essa collegata.
2. Con tutti i componenti del sistema spenti, inserite la cassetta nel registratore.
3. Accendete il vostro video e qualsiasi altra periferica.
4. Se il programma deve essere caricato dal BASIC, premete **[Power]** per accendere la vostra console. Quando il promt del BASIC ready appare, digitate CLOAD e premete **[Return]** sulla tastiera.
   Se il programma non richiede il basic per girare, tenete premuto **[Start]**, quindi premete **[Power]** per accendere la vostra console.
5. Dopo aver sentito un beep, premete il tasto **[Play]** sul registratore, quindi premete il tasto **[Return]** sulla tastiera. Il programma verrà caricato.
6. Se il programma è un programma BASIC, il promt ready apparirà. Digitate RUN per lanciare il programma.

Capire la vostra tastiera

La tastiera XE è come avere tre tastiere in una. Lavora come una convenzionale macchina da scrivere, ma può' essere utilizzata anche per visualizzare caratteri grafici e caratteri di lingue internazionali.

Come una convenzionale tastiera per macchina da scrivere, le lettere, i numeri, la punteggiatura, i tasti **[Schift]**, **[Tab]**, **[Caps]**, e **[Barra spaziatrice]** lavorano allo stesso modo di una macchina da scrivere.

Come caratteri grafici della tastiera, potete visualizzare caratteri decorativi sullo schermo.

Come caratteri internazionali, potete visualizzare caratteri utilizati in lingue straniere.

Inoltre, la tastiera XE ha un numero di tasti speciali per controllare l'XE e i programmi che vengono lanciati in esso. Le funzioni di ogni tasto variano da programma a programma, il seguente paragrafo descrive come questi tasti sono utilizzati nell'ATARI BASIC, dove applicabili, come sono tipicamente usati negli altri programmi. Fate riferimento al manuale che viene fornito con ogni programma per ulteriori informazioni sulle funzioni di questi tasti.

Alcuni di questi tasti sono usati assieme (premendoli simultaneamente) per aggiungere ulteriori funzioni senza incrementare le dimensioni della tastiera.

Fate riferimento al **Capitolo 2, Editare un programma BASIC, con la Tastiera ATARI XE**, sull'utilizzo dei tasti speciali da editare nei programmi ATARI BASIC.

Nota: molti tasti sulla tastiera XE hanno una ripetizione automatica quando vengono premuti per piu' di mezzo secondo.

[Help]
Alcune volte utilizzato per visualizzare istruzioni sull'attuale programma che sta caricando.

[Esc]
Spesso utilizzato per uscire da un menù ad un'altro dentro un programma.

[Delete Bk Sp]
In molti programmi, incluso il BASIC ATARI, cancella il carattere alla sinistra del cursore e muove il cursore di uno spazio sulla sinistra. Non chiude lo spazio lasciato dalla cancellazione.

[Break]
Solitamente interrompe l'attuale attività del computer.

[Return]
Dice al computer che voi avete digitato o editato una linea di testo o una risposta ad un programma.

[Caps]
Shifta tra le lettere maiuscole e minuscole di un programma.

Alcune volte utilizzato con il tasto **[Shift]** per cambiare i modi della tastiera.

Nell'ATARI BASIC esce dai caratteri grafici.

Il ◩ Tasto Inverso Video

Attiva o disattiva la visualizzazione in video inverso. Il video inverso riversa i colori (o bianco e nero) utilizzati sull'inverso video.

[Control]
Sempre utilizzato con altri tasti per permettere funzioni speciali. Nell'ATARI BASIC, stampa caratteri grafici quando utilizzato con i tasti alfabetici.

[Control] [1]
Spesso utilizzato nei linguaggi di programmazione e far ripartire lo scrolling del video.

[Control] [2]
Fa suonare il vostro XE. Il vostro monitor o l'altoparlante del televisore deve essere azionato perche' si possa udire il suono.

[Control] [3]
Produce, in fase di lettura della tastiera, una risposta (EOF) fine del file (usato da programmatori esperti).

[Control] [Insert]
Inserisce uno spazio in molti programmi incluso l'ATARI BASIC.

[Control] [Delete Bk Sp]
Cancella il carattere sotto il cursore e sposta i rimanenti caratteri di una linea per chiudere lo spazio vuoto.

[Control] [↑]
[Control] [↓]
[Control] [←]
[Control] [→]
Muove il cursore in alto, in basso, a sinistra e a destra.

[Control] [Caps]
Blocca il computer nel modo Control. Questo è comodo quando desiderate digitare una serie di tasti utilizzati con il tasto **[Control]**.

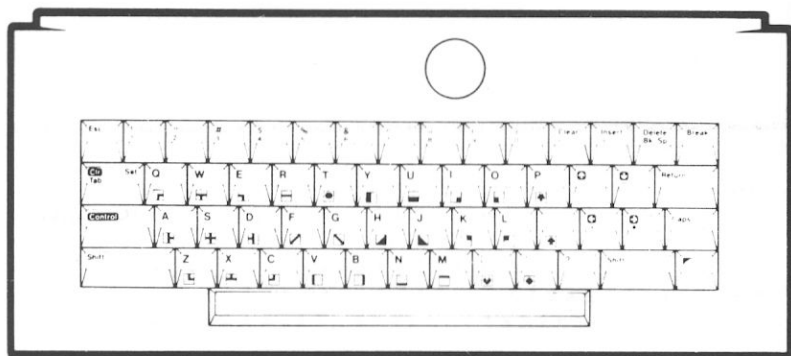
[Shift]
Utilizzato in unione coi tasti letterali per digitare lettere maiuscole.

[Shift] [Insert]
Inserisce uno spazio bianco in un programma ATARI BASIC.

[Shift] [Delete Bk Sp]
Cancella una linea da un programma ATARI BASIC.

[Shift] [Caps]
Blocca il computer nella modalità maiuscola per i caratteri letterali.

Aggiungere Caratteri Grafici Atari

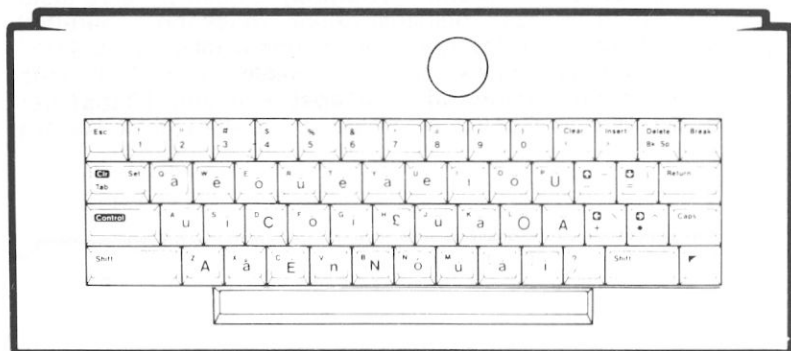
La tastiera ATARI XE ha 29 caratteri grafici residenti. Potete far lavorare la vostra immaginazione e creare anche dei lavori artistici, degli schermi un po' particolari, delle bordature del testo, o altre creazioni grafiche. Per visualizzare i caratteri grafici sullo schermo, premere **[Control]** e qualsiasi altro tasto mostrato sotto. (I tasti nell'illustrazione mostrano i caratteri grafici di fronte alla tastiera). Se intendete utilizzare ulteriori caratteri, potrebbe essere più conveniente bloccare il modo Control premendo **[Control]** e **[Caps]**. Premete **[Caps]** per uscire dal Blocco Control e tornare ai caratteri della tastiera standard.

Aggiungere un Set di Caratteri Internazionali

Quando digitate pagine in lingua straniera, solitamente dovete tornare sulle lettere e laboriosamente aggiungere accenti, cediglie, e altri caratteri particolari. L'ATARI XE vi permette di digitare documenti in lingua straniera facilmente perchè fornisce direttamente questi caratteri.

I caratteri internazionali sono disponibili quando utilizzate l'ATARI BASIC. Il diagramma della tastiera sulla pagina seguente mostra i caratteri internazionali che sono associati con i tasti. Per attivare i caratteri internazionali, digitate l'istruzione qui mostrata e premete **[Return]**

POKE 756,204

Premendo **[Control]** e uno dei tasti mostrati sotto produrrà un carattere internazionale al posto di un carattere grafico. Per tornare alla modalità caratteri grafici e all'utilità della tastiera normale, digitate la seguente linea e premete **[RETURN]**:

POKE 756,224

CAPITOLO 2
EDITARE PROGRAMMI BASIC CON LA TASTIERA ATARI XE

Non importa se sapete scrivere a macchina bene, quello di cui avrete bisogno è familiarizzare con le speciali funzioni della tastiera ATARI XE prima di cominciare a editare con il BASIC ATARI.

Funzioni di Auto Ripetizione

Iniziate digitando la lettera A:

A

Continuate tenendo premuto il tasto **[A]** e guardate la linea di A apparire. Quando una linea è piena il cursore automaticamente va alla riga successiva. Non c'è bisogno di premere **[Return]**.

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

State utilizzando la funzione di autoripetizione della tastiera ATARI XE. Molti tasti, incluso **[Barra Spaziatrice]**, hanno l'autoripetizione. Udite un suono quando le A hanno completato la terza linea? Questo suono è un avvertimento ed è un'istruzione residente dell'ATARI BASIC ed indica che le linee sono troppo lunghe. Una linea d'istruzioni non può superare le tre linee di lunghezza.

Messaggi di Errore

Trovate il tasto **[Return]** e premetelo. Dovreste vedere la parola ERROR sullo schermo, seguita dalle tre linee di A che avete digitato. Il vostro computer è ora interattivo con voi. Vi sta dicendo che non ha capito cosa avete digitato perchè le linee di A non sono parte del linguaggio BASIC.
Pulite lo schermo premendo **[Return]** fin quando il messaggio d'errore non apparirà più. Per evitare i messaggi d'errore durante il proseguimento dell'istruzione di editazione, non premete **[Return]** fino a quando non vi verrà detto.

Maiuscolo e minuscolo

Per ottenere una A minuscola, premete **[Caps]** una volta e tenete premuto **[A]**. Dovreste vedere questo:

aaaaaaa

Per ritornare alle lettere maiuscole, premete **[Caps]** e digitate altre A. Dovreste vedere questo:

AAAAAAA

Cercate di digitare una parola che comincia per A, come ATARI. Digitate le seguenti parole, passando dalle lettere maiuscole alle minuscole con il tasto **[Caps]**:

ATARI XE atari xe

A differenza della macchina da scrivere, il computer ha tasti separati per controllare le maiuscole e lo shifting. In entrambe le minuscole e maiuscole, il simbolo è sempre quello mostrato nella parte bassa del tasto. Per ottenere il simbolo nella parte media del tasto, utilizzate il tasto **[Shift]**. Ci sono due tasti **[Shift]** sulla tastiera. Potete usarli alternativamente.

Utilizzate **[Caps]**, **[Shif]**, e **[1]**, cercate di digitare queste:

! ! ! ATARI XE ! ! ! ! ! ! atari xe ! ! !

Sperimentate con differenti parole, lettere e punteggiature.

Simboli Grafici

Molti tasti hanno due o tre simboli. Ogni tasto letterale ha una lettera sulla parte superiore e un simbolo grafico sulla parte frontale. Alcuni altri tasti hanno tre simboli o parole tutti sulla parte superiore. Una funzione dei tasti è attivata attraverso la pressione del tasto, altri premendo **[Shift]** e il tasto, e terzo premendo **[Control]** e il tasto.
I simboli grafici sono prodotti premendo **[Control]** e il tasto.

Per digitare un simbolo grafico (il simbolo sul lato frontale del tasto), usate il tasto **[Control]** sulla sinistra della tastiera. Prima premete **[Control]**. Mentre state premendo il tasto **[Control]**, premete un simbolo grafico. Quindi otterrete entrambi i tasti.

Tenete premuto **[Control]** e digitate ATARI. Dovreste vedere questo:

Solo cinque caratteri appaiono sullo schermo. Se usate il tasto **[Control]** quando premete un tasto numerico, nessun simbolo grafico appare.

I simboli grafici sono utili nella visualizzazione di disegni grafici, bordi, e semplici lavori artistici. Potete bloccare la tastiera nella modalità simboli grafici tenendo premuto **[Control]** e quindi premendo **[Caps]**. Premendo **[Caps]** solo una volta tornerete alla modalita' minuscolo.

Controllo del Cursore

Il tasto **[Control]** è utilizzato più frequemtemente per direzionare il movimento del cursore. Il cursore è il piccolo quadrato bianco che delimita uno spazio sullo schermo. Trovate il tasto **[Freccia verso l'alto]** vicino alla lettera P. La freccia, come il tasto **[Control]**, è disegnata in bianco. Questa marcatura in bianco indica che la funzione di freccia è attivata solo utilizzando il tasto **[Control]**. Premendo **[Control]** e quindi **[Freccia verso l'alto]** e guardate il cursore muoversi nella parte alta dello schermo. Quando raggiunge la parte alta dello schermo, il cursore ritorna nella parte bassa dello schermo e ricomincia a muoversi di nuovo. Ora cercate i tasti **[Cursore verso il basso]**, **[Cursore a destra]** e **[Cursore a sinistra]**. Ricordate di utilizzare il tasto [Control].

Pulire lo schermo

Il tasto **[Control]** viene frequentemente utilizzato in unione col tasto **[Clear]** per cancellare tutto sullo schermo. Premete e tenete premuto **[Control]**, quindi premete **[Clear]**. Questa operazione dovrebbe pulire il vostro schermo e far tornare il cursore nell'angolo in alto a sinistra dello schermo stesso. Riprovate.

Ora riempite lo schermo con lettere, parole, numeri e simboli grafici. Questa volta utilizzate **[Shift]** con **[Clear]** per pulire lo schermo. Sia **[Shift] [Clear]** che **[Control] [Clear]** svuotano lo schermo e riportano il cursore nell'angolo in alto a sinistra.

Inserimento

Il tasto **[Control]** è utilizzato con il tasto **[Insert]** per inserire spazi in una linea. Per fare pratica con questa funzione, digitate:

!!!ATARI XE!!!

Posizionate il cursore all'inizio della prima lettera A in ATARI. Tenete premuto **[Control]**, premete **[Insert]** 11 volte. Dovreste vedere questo:

!!!   ATARI XE!!!

Undici spazi bianchi sono stati aggiunti nel mezzo della linea. Questa funzione è molto utile per inserire parole. Utilizzando i tasti di controllo del cursore (le frecce), tornate agli spazi vicino al terzo punto di domanda vicino alla linea. Premete **[Barra spaziatrice]** una volta e digitate QUESTO E' UN negli spazi bianchi come mostrato sotto:

!!!THIS IS AN ATARI XE!!!

Per aggiungere linee bianche, piuttosto che spazi individuali, premete **[Shift]**, quindi premete **[Insert]**. Una intera linea bianca apparirà sullo schermo. Inserite altre linee bianche, ma non così tante da avere uno schermo bianco. Tenete la frase sullo schermo in modo tale da poter procedervi con il prossimo esercizio.

Cancellazione

Utilizzando il tasto **[Control]** con il tasto **[Delete Bk Sp]** vi premette di cancellare in modo facile un inserimento. Posizionate il cursore sulla Q della parola QUESTO. Tenete premuto **[Control]**, premete **[Delete Bk Sp]** 12 volte. Il vostro schermo dovrebbe apparire così:

!!!ATARI XE!!!

Ora sapete come i tasti **[Control]** e **[Delete Bk Sp]** lavorano. Per scoprire cosa fa il tasto **[Delete Bk Sp]** quando premuto, posizionate il cursore sulla prima A e premete **[Delete Bk Sp]** tre volte. Il vostro schermo dovrebbe apparire così:

ATARI XE!!!

Quando utilizzato da solo, il tasto **[Delete Bk Sp]** muove il cursore a sinistra, cancellando durante il suo movimento, ma non chiude gli spazi. Utilizzando il tasto **[Control]** con il tasto **[Delete Bk Sp]** cancellate i caratteri alladestra e chiudete la frase.

La terza funzione del tasto **[Delete Bk Sp]** richiede l'utilizzo del tasto **[Shift]**. Premendo e tenendo premuto **[Shift]** e quindi premendo **[Delete Bk Sp]** cancellate un'intera linea e il cursore torna al margine sinistro. Non importa dove il cursore si posiziona sulla linea quando premete **[Shift] [Delete Bk Sp]**; l'intera linea viene cancellata dallo schermo.

Tabulazioni

Su uno schermo bianco muovete il cursore sul margine sinistro e digitate un asterisco. Premete **[Tab]**. Ogni volta che il cursore si ferma, digitate un asterisco. Dovreste avere sei asterischi spaziati lungo lo schermo come mostrato sotto:

*　　*　　*　　*　　*　　*

Premete **[Tab]** solamente e notate che si ferma alla stessa posizione del primo asterisco. La prima posizione di tabulazione è costituita da cinque spazi dal margine sinistro (una normale indentazione di paragrafo), e i seguenti punti di tabulazione sono otto spazi l'uno dall'altro. Posizionate il cursore nella parte alta del primo asterisco e muovetelo di tre spazi. Premete e tenete premuto **[Shift]** e quindi premete **[Tab]** per attivare la funzione **[Set Tab]**. Muovete il cursore indietro al margine sinistro, quindi premete **[Tab]**. Il cursore salta alla nuova posizione di tabulazione.

Continuate a premere **[Tab]**. Il cursore continua ad andare a tutte le posizioni di tabulazione, in aggiunta a quella nuova. Quando il cursore salta alla linea successiva, ignora la nuova posizione di tabulazione. (ma sulle seguenti linee il cursore andrà alle posizioni di tabulazione, quella nuova e quelle già preselezionate). Riportate il cursore al primo asterisco e premete **[Tab]**. Il nuovo punto di tabulazione esiste ancora.

Riportate il cursore al margine sinistro. Premete **[Tab]** muovetevi al primo punto di tabulazione (tre spazi). Utilizzate **[Control]** con **[Tab]** per attivare la funzione **[Clr Tab]**. Premete **[Tab]** per ottenere la successiva posizione di tabulazione e pulirla. Muovete il cursore indietro al margine sinistro della stessa linea e premete **[Tab]** solamente. Il cursore dovrebbe saltare di due posizioni di tabulazione. Continuate a premere **[Tab]** fino a che il cursore salta alla linea successiva. **[Clr Tab]** non pulisce la seconda posizione di tabulazione in questa linea. (Comunque, entrambe le posizioni di tabulazione sono state ripulite dalla seguente linea).

Video Inverso

Digitate la parola ATARI. Trovate il tasto ◩ **[video inverso]** e premetelo una sola volta. Digitate ATARI di nuovo. Premete **[video inverso]** ancora e digitate ATARI nuovamente. Il vostro schermo dovrebbe risultare il seguente:

ATARI ATARI ATARI

Il video inverso crea lettere blu su sfondo bianco, l'inverso dei colori normali. Questa funzione è molto utile per visualizzare lettere in un vostro programma. Solo un tocco del tasto **[video inverso]** cambia il modo di visualizzazione delle lettere.

Tasti con Funzioni Diverse

Un'altro importante tasto è il tasto Escape **[Esc]**. Quando lo premete una volta non succede nulla. Quando lo premete due volte o più, sullo schermo appare il simbolo ⇤ . Premete **[Return]** e provate di nuovo. Nei successivi capitoli avrete bisogno di utilizzare il tasto **[Esc]**.

Il tasto **[Break]** si trova nelle parte alta a destra della tastiera. Quando premete questo tasto, il cursore salta una linea e muove il margine sinistro. Nel **Capitolo 7, Produzione di Suoni e Grafici con il Basic**, imparerete come usare il tasto [Break].

Quando premete i tasti **[Help] [Start] [Select]** e **[Option]**, non succede nulla. Questi tasti sono programmabili e spesso hanno funzioni in programmi software.

Quando premete il tasto **[Reset]**, lo schermo ritorna nero per un secondo o due, e il promt Ready apparirà. Il tasto **[Reset]** riattiva il sistema. Dovreste usare questo tasto in momenti sporadici dato che, in molti programmi, le informazioni che avete inserito verranno perdute.

CAPITOLO 3
SCRIVERE UN SEMPLICE PROGRAMMA BASIC

Una volta presa confidenza con la tastiera del computer, diventa facile scrivere il vostro primo programma. Per cominciare, pulite lo schermo e assicuratevi che il cursore sia sul margine sinistro.

NEW: Pulire la Memoria del Computer

Digitate la parola NEW, quindi premete **[Return]**:

NEW

NEW dice al computer di prepararsi una nuova sezione di istruzioni cancellando tutte le vecchie istruzioni che potrebbero essere nella nella memoria del computer.

LIST: Controllo della Memoria del Computer

Per assicurarsi che nella memoria del computer non ci sia più nulla, chiedete al computer stesso di listarvi qualsiasi istruzione che potrebbe aver memorizzato. Digitate su una linea LIST e premete **[Return]**:

LIST

Se digitate NEW correttamente, nessun altro promt all'infuori di Ready apparirà sul vostro schermo. Ora potete cominciare un nuovo programma. Digitate la prima linea d'istruzioni nel computer. Digitate la linea esattamente come appare sotto e premete **[Return]** dopo l'ultimo apice:

10 PRINT "I HEARD OF A POET NAMED SAM"

Tutte le linee d'istruzione nei programmi Basic sono numerate. Quando digitate questo programma formato da una linea, assicuratevi che 1 e 0 nel numero 10 siano numeri, non lettere. Se utilizzate lettere al posto di numeri, avrete un messaggio d'errore.

Una linea di istruzioni numerata in un programma può essere più lunga di una linea sullo schermo. Quando il cursore oltrepassa lo spazio di una linea, automaticamente salta a una linea successiva. Dovreste premere **[Return]** solo alla fine di una linea di istruzione per dire al computer che avete digitato le istruzioni e che dovrebbero essere memorizzate nella sua memoria. Non succede nulla quando premete **[Return]**; il cursore ritorna al margine sinistro così potete cominciare un'altra linea del programma.

RUN: Esecuzione di Istruzioni

Per fare in modo che il computer esegua il vostro programma, dovete digitare RUN. Il comando RUN dice al computer di estrarre le proprie istruzioni. Digitate RUN e premete **[Return]** per vedere cosa accade:

RUN
I HEARD OF A POET NAMED SAM

La prima e unica istruzione del programma, la linea 10, serve per stampare la parola all'interno degli apici. Pulite lo schermo, digitate RUN ancora e premete **[Return]**. Il computer esegue la sua istruzione di nuovo e stampa:

I HEARD OF A POET NAMED SAM

Anche se l'istruzione non è più presente sullo schermo il computer ricorda cosa fare. Il vostro programma è memorizzato in RAM (Memoria ad Accesso Casuale), la sezione programmabile della memoria del computer. Quando digitate LIST il computer mostra sullo schermo tutte le funzioni memorizzate nella porzione di RAM della sua memoria. Digitate LIST. Il vostro schermo dovrebbe essere il seguente:

LIST
10 PRINT "I HEARD OF A POET NAMED SAM"

Se il vostro schermo è differente, potreste avere dimenticato di premere **[Return]** alla fine di ogni inserimento o digitato LIST su una linea a se stante. Digitate la linea sottostante, quindi date il comando RUN:

20 PRINT "I MET HIM ONE DAY, AND TO MY DISMAY"
RUN

Le parole chiuse tra apici in entrambe le linee del programma appaiono sullo schermo. Digitate LIST per vedere le istruzioni che il computer ha memorizzato in RAM. Entrambe le linee 10 e 20 appaiono.

NUMERAZIONE DI LINEE: Creazione dell'Ordine

Ogni linea d'istruzione in un programma BASIC deve avere un numero di partenza. I numeri sono chiamati "numeri di linea". Il computer esegue le istruzioni, cominciando dal numero più piccolo e continuando attraverso il programma fino a che le istruzioni sono terminate. Solitamente la procedura è quella di numerare per dieci in modo tale da avere abbastanza numeri per l'inserimento di linee successive. Ora cercate di inserire una linea. Aggiungete la linea 15 (al seguente esempio) e istruite il computer per lanciare il programma. Il vostro schermo dovrebbe risultare come il seguente programma:

15 PRINT "WHOSE POEMS WERE THE TALK OF THE LAND."
RUN
I HEARD OF A POET NAMED SAM
WHOSE POEMS WERE THE TALK OF THE LAND.
I MET HIM ONE DAY, AND TO MY DISMAY,

Il computer inserisce automaticamente la linea 15 tra la linea 10 e 20. Scrivete un'altra linea:

30 PRINT "HIS BRAINS WERE SILICON-SAND."
RUN
LIST
I comandi RUN e LIST stampano sullo schermo tutte le istruzioni racchiuse nel comando PRINT.

MESSAGGI di ERRORE: Il Computer parla per "Non Ho Capito"

PRINT dice semplicemente al computer di stampare tutto quello che è contenuto tra gli apici sullo schermo. Il computer non si interessa sul tipo di parole o simboli all'interno degli apici; le parole non hanno bisogno di essere scritte correttamente o avere un senso. Provate ora le istruzioni qui scritte: anche quando gli apici racchiudono una frase senza senso contenente parole errate, il computer fa quello che gli è stato detto. In ogni caso, cercate di scrivere PRINT in modo sbagliato come mostrato qui sotto per vedere cosa succede:

40 PRINT "AYE SAY HYE; U SAY BI."
RUN
Il computer vi segnala un messaggio d'errore. Il computer verifica solo quelle istruzioni che sono al di fuori degli apici perchè queste istruzioni vengono interpretate dal computer. Le istruzioni tra gli apici sono inserite da voi, così il computer le copia esattamente. Muovetevi in una linea bianca, ma non pulite lo schermo. Lanciate il programma per vedere cosa succede.

50 PRIMT "I SAY HI; YOU SAY BYE."

Il messaggio d'errore 17 appare alla linea 50, la linea in cui avete intenzionalmente digitato in modo erroneo PRINT. Il messaggio d'errore 17 è un "errore di sintassi". Indica che le istruzioni sono indecifrabili per il computer. (Per una completa lista di messaggi d'errore, vedi **Appendice D**).

Ci sono diversi modi di correggere un messaggio d'Errore. La soluzione più facile è quella di muovere il cursore sulla linea che contiene l'errore digitato. Posizionate il cursore sulla M di PRIMT e cambiatela in N. Premete **[Return]**.(In questo caso, potete premere **[Return]** indipendentemente dal cursore sulla linea, fino a che è nel mezzo della parola PRINT.) Nessun nuovo messaggio d'Errore appare questa volta. Pulite lo schermo e lanciate il programma. Lo schermo non dovrebbe presentare nessun messaggio d'errore.

Un altro modo per correggere un messaggio d'errore è quello di cancellare la linea errata. Per fare pratica con questa tecnica, digitate un'altra linea che contiene un errore intenzionale. Questa volta omettete l'apice nell'istruzione PRINT, quindi lanciate e listate il programma:

60 PRINT I ONCE HAD PROGRAM CALLED BOZON
RUN
LIST

Un messaggio d'errore appare quando premete **[Return]** e quando cercate di lanciare e listare il programma. Per cancellare la linea errata, semplicemente digitate il numero della linea e premete **[Return]**:

60
RUN
LIST

Ora il programma può essere lanciato e listato senza errori, anche la linea 60 non contiene alcuna istruzione. La linea I ADSESSO HO UN PROGRAMMA CHIAMATO BOZON è stata cancellata. Digitate il numero di linea e premete **[Return]** cancellando interamente una linea dalla memoria del computer. Digitate la linea correttamente come mostrato sotto:

60 PRINT "I ONCE HAD PROGRAM CALLED BOZON"
RUN

PRINT: Creazione di Linee Bianche

Inserire una linea bianca dopo uno scritto aiuterebbe al testo inserito ad essere più leggibile. Digitate le seguenti istruzioni per creare una linea bianca tra le linee 30 e 40:

35 PRINT
RUN
LIST

Quando nulla segue il comando PRINT, il computer crea una linea bianca. Inserite un'altra linea bianca tra le linee 50 e 60. Utilizzate il numero 55 per numerare la linea e digitate solo la parola PRINT dopo il numero.

?: Abbreviazione di Print

Potete risparmiare tempo sostituendo un punto di domanda (?) a PRINT. Digitate la seguente linea di programma:

70 ? "THAT RAN FROM DUSK UNTIL DAWN."
RUN
LIST

Il programma gira allo stesso modo che con PRINT. Il punto di domanda è solo una conveniente abbreviazione. Per chiarezza vi diciamo subito che potete utilizzare il punto di domanda al posto di PRINT, in tutti i successivi esercizi.

Lunghezza Logica della Linea

Alcune volte gli apici contengono troppi caratteri per essere inseriti in una o due linee. Assicuratevi che l'audio del vostro televisore o del vostro monitor sia udibile prima di inserire il seguente esempio:

80 PRINT "IT WOULDN'T RESPOND TO ESCAPE, BREAK, CONTROL, OR LIST, AND IT WAS STILL RUNNING WHEN I TURNED OFF THE SWITCH."

Quando il cursore raggiunge la terza linea udirete un suono. Questo suono indica che vi siete avvicinati alla massima lunghezza di una linea d'istruzione. Una linea d'istruzione non può aver lunghezza superiore a 3 linee dello schermo. Questo limite viene chiamato "linea logica". (Ora potete anche abbassare il volume.)

Lo Schermo di Visualizzazione

Le parole sono spesso rotte in diversi punti quando il cursore raggiunge la fine di una linea di schermo. Anche lo spazio tra le parole quando digitate una parola in una linea di programma è differente dallo spazio quando il computer lancia il programma. Per far combaciare questi problemi, determinate la struttura della linea e digitate separate istruzioni PRINT per ogni linea. Ridigitate le istruzioni nella linea 80 in modo tale che il listato appaia così:

80 PRINT "IT WOULDN'T RESPOND"
90 PRINT "TO ESCAPE, BREAK, CONTROL, OR LIST,"
100 PRINT "AND IT WAS STILL RUNNING"
110 PRINT "WHEN I TURNED OFF THE SWITCH"
RUN
LIST

Stampare Simboli Grafici

Potete anche utilizzare simboli grafici nell'istruzione PRINT per produrre semplici decorazioni. Per disinserire il listato, digitate le linee sottostanti. Utilizzate **[Control] [J]** e **[Control] [H]** per creare i grafici:

58 PRINT "◣"
115 PRINT "◢"

Print "↖" : Pulire lo Schermo

Potete migliorare il vostro programma assicurandovi che lo schermo sia pulito quando iniziate. Digitate un numero di linea, PRINT, e il primo apice. Premete **[Esc]** una volta. Quindi premete sia **[Shif]** e **[Clear]** o **[Control]** e **[Clear]**. Una freccia verso sinistra appare sullo schermo. Digitate un'altro apice e premete **[Return]**. Quindi lanciate e listate il programma:

5 PRINT "↖"
RUN
LIST

[Control] [1]: Fermare la Visualizzazione sullo Schermo

Ora il programma sembra migliore, ma è troppo lungo per tutte le linee apparse assieme sullo schermo. Quando il computer lista il programma, potete fermare le linee che si muovono verso l'alto premendo i tasti **[Control]** e **[1]**. Digitate LIST. Utilizzate due dita della vostra mano sinistra e premete **[Control]** e **[1]** e un dito della vostra mano destra per prememere **[Return]**. **[Control]** e **[1]** etrambi cominciano e fermano la funzione LIST.

CAPITOLO 4
CREARE UN LOOP INTERATIVO IN BASIC

I loop dicono al computer di tornare e ripetere istruzioni automaticamente in un programma. Il comando GOTO vi da la possibilità di non ripetere all'infinito la digitazione di alcune linee. I comandi DIM e IMPUT vi permettono di interagire col vostro computer sulla base della domanda e risposta. Inserite questi tre comandi assieme-GOTO, DIM, INPUT-vi permette di avere interessanti conversazioni col vostro computer.

GOTO: Una Mappa del Computer

Il loop più semplice del computer è il loop GOTO. GOTO è sempre seguito da un numero di linea che dice dove saltare alla prossima istruzione. Avete bisogno solo di due comandi per creare un loop. Digitate il programma sottostante per produrre un loop infinito:

```
NEW
110 PRINT "CONGRATULATIONS !"
120 GOTO 110
RUN
```

Per fermare questo loop infinito, spegnete il computer o utilizzate il tasto **[BREAK]**, uno dei seguenti messaggi appare:

STOPPED AT LINE 110

o

STOPPED AT LINE 120

Il computer vi sta dicendo dov'era quando ha ricevuto il comando stop.

Virgola: Un Tabulatore

Il loop GOTO vi risparmia una grossa quantità di lavoro con solo due righe di lavoro. Per avere facilitazioni dal programma, listate il vostro programma, posizionate il cursore nello spazio successivo all'ultimo apice, inserite una virgola, e premete **[Return]**. Lanciate il programma e guardate gli effetti speciali:

```
LIST
110 PRINT "CONGRATULATIONS !",
120 GOTO 110
RUN
```

La virgola agisce come un tabulatore. Ogni volta che il computer va alla linea successiva e stampa CONGRATULAZIONI! muove alla successiva posizione di tabulazione. Il risultato ha un effetto molto interessante. Ricordate che per fermare il loop dovete premere il tasto **[Break]**.

Punto e Virgola: Una Particolarità del Computer

Un punto e virgola produce un'altro divertente effetto. Listate il programma, cambiate la virgola nella linea 110 con
un punto e virgola, premete **[Return]** e lanciate il programma:

```
LIST
110 PRINT "CONGRATULATIONS !";
120 GOTO 110
RUN
```

Il punto e virgola unisce le istruzioni PRINT assieme senza spazi tra loro. Mettete alcuni spazi tra le parole, tornate ed editate la linea 110 in modo tale da apparire come segue:

```
110 PRINT "CONGRATULATIONS !
RUN
```

Due Punti: Un Separatore

Il due punti è un separatore. Permette a due istruzioni di essere susseguenti su una linea. Cambiate il punto e virgola nella linea 110 con un due punti e aggiungete il comando PRINT come mostrato sotto:

```
110 PRINT "CONGRATULATIONS !": PRINT "YOU JUST
WON THE LOTTERY."
RUN
```

Ora che state progredendo in abilità come programmatori, avete anche bisogno di fare un'operazione importante, cioè lasciare più spazio nella memoria del computer. Più comandi su una linea separati da un due punti è uno dei modi per Il computer risponde con un numero. Riprogrammate la linea 110 in modo tale che le due istruzioni PRINT siano in due linee di programma:

```
PRINT FRE (0)
```

salvaguardare bytes liberi di memoria RAM. (Un byte è pari a un carattere) Per vedere quanta memoria rimane digitate la seguente istruzione:

```
110 PRINT "CONGRATULATIONS !"
115 PRINT "YOU JUST WON THE LOTTERY."
PRINT FRE(0)
```

Confrontate i due numeri dei bytes disponibili. Il secondo numero è due o tre volte inferiore al primo. Cio significa semplicemente che è più importante cominciare a programmare che conservare la memoria del computer, i numeri diprogramma di questo capitolo conterranno solo un'istruzione per linea. Un'eccezione verrà fatta con l'istruzione PRINT che inserisce una linea bianca tra segmenti di programma. Digitate la nuova linea che vedete sotto per vederne gli effetti :

```
110 PRINT: PRINT "CONGRATULATIONS !"
RUN
```

DIM e INPUT: Dimensionamento e Inserimento di Variabili Stringa

XE deve essere programmato per rispondere a una domanda. Potete utilizare un comando PRINT per fare una domanda e il comando INPUT per inserire una risposta nel computer. In ogni caso, quando date una risposta al computer, il computer deve conoscere dove inserirla. La risposta viene inserita in una "variabile" nella memoria RAM. Se la risposta e' composta da lettere, numeri, o entrambi, viene chiamata una "variabile stringa". Il vostro ATARI XE ha bisogno di sapere quanto spazio avete bisogno per una risposta in modo tale da riservare spazio per essa. Questo processo viene chiamato "dimensionamento di variabili stringa".

Il comando DIM (dimensionamento) è sempre accompagnato dal comando INPUT per variabili stringa dato che DIM determina la dimensione della risposta. Per le variabili la dimensione viene riferita al numero di caratteri, incluso gli spazi, che sono necessari. Dovete dire al computer il numero massimo di spazi che la risposta occupa.

Cambiate il programma loop in uno che fa una domanda e aspetta la risposta. Non c'e' bisogno di riscrivere il programma; inserite solo le nuove linee- linea 10, 120, 130 e 140-come mostrato sotto. (Digitando la nuova linea 120 automaticamente cancellate la vecchia linea 120.)

```
10 DIM ANSWER$ (100)
110 PRINT: PRINT "CONGRATULATIONS !"
115 PRINT "YOU JUST WON THE LOTTERY."
120 PRINT: PRINT "HOW DOES THAT MAKE YOU FEEL?"
130 INPUT ANSWER$
140 PRINT "I THOUGHT YOU WOULD SAY THAT."
RUN
```

La linea 10 dice al computer di salvare abbastanza spazio per la risposta che ha un massimo di 100 caratteri. La variabile in questo programma è stata chiamata ANSWER. La variabile memorizza lettere e numeri, cioè è una variabile stringa. Le variabili stringa sono caratterizzate dal segno dollaro dopo l'ultima lettera della variabile.

La linea 130 vi permette di inserire una risposta. Quando lanciate il programma, il computer visualizza la domanda sullo schermo, e voi potete digitare la vostra risposta sullo schermo. Questa memoria è memorizzata nella variabile stringa chiamata ANSWER$. Se l'istruzione DIM, linea 10, è stata omessa, un messaggio d'errore verrà visualizzato, e il comando INPUT non funzionerà.

?: Cortesia dell'Input

Lanciate il programma di nuovo. Due punti di domanda appariranno sullo schermo. Il secondo punto di domanda sarà sulla linea successiva al margine sinistro. Listate il vostro programma e notate che avete digitato solo un punto di domanda nel programma. Il comando INPUT sempre inserisce un punto di domanda per voi. Digitate la variazione della linea 120 mostrata sotto:

120 PRINT "HOW DOES THAT MAKE YOU FEEL"

Lanciate il programma e digitate la vostra risposta quando il computer vi fa la domanda. Ora solo un punto di domanda appare, e la vostra risposta segue immediatamente il punto di domanda nella stessa linea. Create altri dialoghi dimensionando più variabili stringa e inserendo più istruzioni INPUT. Le istruzioni DIM dovrebbero essere messe all'inizio del programma:

20 DIM DATE$ (25)
140 PRINT: PRINT " WHEN WOULDYOU LIKE TO COME AND PICK UP YOUR PRIZE";
150 INPUT DATE$
RUN

Le Variabili Stringa nell'Istruzione PRINT

Il programma ora vi fa due domande, ma non vi risponde all'ultima. Per avere una risposta potete inserire la variabile stringa nell'istruzione PRINT nel modo seguente:

160 PRINT "I'M SORRY, BUT OUR OFFICES ARE ALWAYS CLOSED ON"; DATE$; ". TOO BAD!"

Il punto e virgola unisce le variabili stringa tra due frasi nei punti di domanda. Lanciate il programma. Le parole non sono spaziate correttamente, confrontate la vostra linea con quella sottostante. Probabilmente avete tralasciato uno spazio dopo la N di ON o dimenticato una virgola e lo spazio prima di TOO BAD! Questi spazi sono importanti. Fate pratica con altre variabili stringa in input:

```
30 DIM NAME$ (1)
170 PRINT "BY THE WAY, WHAT IS YOUR NAME";
180 INPUT  NAME$
190 PRINT "WELL, ";NAME$; ", I BET YOU WOULD LIKE
TO KNOW HOW MUCH YOU WON, FIRST YOU HAVE TO
ANSWER A QUESTION."
```

Lanciate il programma. Anche se avete digitato il nome completo il computer stampa solo l'iniziale. Ciò accade perchè l'area dimensionata nella memoria RAM per il nome è troppo piccola. Molti nomi di persone sono più lunghi di un solo carattere. Cambiate la linea 30 con un numero di spazi più ragionevole e lanciate il programma:

```
30 DIM NAME$ (25)
RUN
```

Inserimento di variabili numeriche

Fino a questo momento avete lavorato con variabili stringa alfanumeriche-variabili composte da lettere, numeri o entrambi. Cioè, il computer accetta il nome R2-D2 o 007 come una variabile stringa. In ogni caso, il nome numerico dovrebbe essere usato solo come un nome, non come un nome in problemi matematici. Ora cercate alcune variabili numeriche che possono essere usate in calcoli matematici. Variabili numeriche non hanno bisogno del comando DIM o del segno $. Inserite le seguenti linee di programma:

```
200 PRINT: PRINT "HOW OLD ARE YOU";
210 INPUT AGE
220 PRIZE=AGE*1000
230 PRINT: PRINT "YOU HAVE JUST WON $"; PRIZE;
"FORM THE LOTTERY. YOU CAN COLLECT DURING
OFFICE HOURS."
```

In questo programma, l'età inserita viene memorizzata nella variabile numerica chiamata AGE. La linea 220 crea un'altra variabile chiamata PRIZE. La linea 220 permette al calcolatore interno al computer di calcolare il prezzo che è 1000 $ moltiplicato per l'età del vincitore. (Per il computer * significa moltiplicazione). Il programma effettua il calcolo matematico per voi, memorizza il risultato in PRIZE. La linea 230, che posiziona la variabile numerica all'interno dell'informazione PRINT nello stesso modo come fosse una variabile stringa,vi dice qual'è la

risposta.

Loop in Input

Per ripetere la vostra conversazione col computer, aggiungete un comando di loop al programma. Un'istruzione GOTO farà ripetere al computer il programma dall'inizio. Per la leggibilità del programma, utilizzate un'istruzione REM per mostrare dove la porzione principale del programma ha inizio.Una istruzione REM funziona come una nota per il programmatore. Il computer ignora i comandi REM, ma li stampa quando listate il programma:

```
100 REM *** CONVERSATION LOOP ***
240 GOTO 100
```

Il compuetr deve ritornare alla linea 100, piuttosto che alla linea 10, dato che non va oltre il comando DIM per dimensionare la variabile. Se il loop oltrepassa il messaggio DIM, riceverete un messaggio d'errore.

CAPITOLO 5
UTILIZZO DI NUMERI CASUALI E FUNZIONI MATEMATICHE CON IL BASIC

Inizialmente i computer furono realizzati per effettuare operazioni numeriche veloci e semplici. Per sfruttare al massimo le capacità del computer nei calcoli matematici in pochi millisecondi, dovete conoscere come parlare a un computer.

Numeri

Digitate NEW per cancellare il precedente comando; quindi digitate l'istruzione e premete **[Return]**:

PRINT 10

Il computer dovrebbe stampare il numero 10. Assicuratevi di utilizzare i numeri 1 e 0, non lettere. Fate pratica stampando i seguenti numeri:

PRINT 1000000000
PRINT -100000000

Utilizzate il segno meno (-) sul tasto **[Freccia Verso l'Alto]** per indicare numeri negativi. Non utilizzate virgole nei numeri. Digitate le istruzioni sottostanti per vedere cosa accade quando utilizzate le virgole:

PRINT 9,876,543,210
PRINT 9, 876, 543, 210

In entrambi gli esempi, il computer interpreta le virgole come separatori in una serie di numeri. Egli spazia i numeri attraverso lo schermo considerando le sue posizioni di tabulazione prestabilite. Per il computer, il 9 non è 9 bilioni, ma solamente il numero 9 seguito da una serie di altri numeri. (ndt: si prenda in considerazione la sintassi americana di formattamento dei numeri!)

Notazione Scientifica

Il computer può non capire virgole quando stampa numeri, ma capisce gli esponenti. Spesso traduce automaticamente una grossa quantità di numeri in forma esponenziale. Digitate i numeri sottostanti:

PRINT 99999999999
PRINT 55555555555
PRINT 111111111111
PRINT -11111111111
PRINT -98765432112

I numeri in base alla loro dimensione verranno riscritti dal computer in notazione scientifica. Non importa se non avete capito perfettamente la notazione scientifica dato che non è basilare per la comprensione del computer, o di questo capitolo.

La notazione scientifica esprime una notevole quantità di numeri tra 0 e 10 multipli elevati alla decima. L'esponenziale viene considerato in base 10. Nel seguente esempio E+13 significa che l'esponente è 13:

2.5E+13 =2.5 x 10^{13} =25000000000000

Potete utilizzare esponenti per parlare col vostro computer. Il simbolo sul tasto **[Freccia Verso Destra]** è il simbolo di esponenziale. Dovete usare il tasto **[Shift]** per stamparlo. Esaminate ora i seguenti calcoli:

PRINT 2 ^ 1
PRINT 2 ^ 2
PRINT 2 ^ 3
PRINT 2 ^ 4
PRINT 2 ^ 64

La prima notazione è due elevato alla prima potenza; la seconda, 2 elevato alla seconda; e così via. L'ultima notazione è 2 alla 64esima, che è un numero abbastanza grosso per il computer per essere espresso in notazione scientifica.

Effettivamente l'applicazione della notazione scientifica abbraccia campi molto ristretti. In ogni caso, se dovrete farlo, Ricordate che il computer è capace di operare sia con un tipo di notazione che con un'altra.

Il Computer come un Calcolatore

Il computer può effettuare le stesse funzioni di un calcolatore. Utilizzate il segno più (+) sul tasto **[Freccia a Sinistra]** per digitare l'istruzione sottostante:

PRINT 1+1

Quando premete **[Return]**, il computer immediatamente vi da la risposta, proprio come un calcolatore. Inventate da soli delle addizioni. Inserite numeri grossi e piccoli, e cercate una lunga serie di numeri da addizionare. Effettuate degli esperimenti con alcune varianti.

Utilizzate il segno meno sul tasto **[Freccia Verso L'Alto]** per sottrarre numeri. Provate le tre versioni del problema sottostante:

PRINT 4 - 1
PRINT 4-1
PRINT4-1

La stessa risposta apparirà per ogni esempio appena premete **[Return]**. La spaziatura nei problemi matematici non è importante per il computer. Cercate da soli dei problemi. Realizzate dei lunghi problemi che combinano addizioni e sottrazioni.

Il segno di moltiplicazione -l'asterisco (*)-è situato sul tasto **[Freccia Verso l'Alto]**. Il segno di divisione è lo slash (/) sul tasto **[?]**. Digitate i comandi seguenti:

PRINT 2 * 2
PRINT (2*2)
PRINT 6 / 3
PRINT (6/3)

Il computer non solo capisce l'utilizzo delle parentesi nei problemi matematici, ma necessita di esse quando i problemi diventano complessi. Si noti cosa accade in questo problema con e senza parentesi:

PRINT 3* (2+2)
PRINT 3*2+2

La risposta al primo problema è 12; la risposta al secondo problema è 8. Nel primo problema, il computer prima somma 2 col 2, poi moltiplica per 3 per arrivare a 12. Nel secondo problema, il computer prima moltiplica il 2 col 3, poi somma il 2 per arrivare a 8. Tutte le volte che il computer incontra parentesi in un problema matematico, effettua il calcolo all'interno delle parentesi come prima operazione e quindi termina il resto del calcolo.

Cercate ora di risolvere i problemi sottostanti per scoprire qualche altro interessante fatto su come lavora il vostro computer. Osservate se potete predire la risposta prima di premere **[Return]**:

```
PRINT (2+2)*3
PRINT 2+2*3
```

Nel primo problema, il computer effettua come prima operazione il calcolo all'interno delle parentesi. Nel secondo problema il computer effettua come prima operazione la moltiplicazione, quindi l'addizione. Il computer esegue queste funzioni matematiche in ordine di specifiche prestabilite: prima, calcola le operazione all'interno delle parentesi; in secondo luogo, le funzioni esponenziali; terzo, le funzioni di moltiplicazione e divisione come appaiono nel problema da sinistra verso destra; e infine, funzioni di sottrazione e addizione da sinistra verso destra. Le regole sono elencate nella seguente tabella:

Ordine di Esecuzione Matematica

1.	()	Calcoli tra parentesi	
2.	^	Funzione esponenziale	
3.	*	Moltiplicazione	In ordine di apparizione
	/	Divisione	da sinistra a destra
4.	+	Addizione	In ordine di apparizione
	-	Sottrazione	da sinistra a destra

Numeri Casuali

Il computer può effettuare alte funzioni che un calcolatore solitamente non può fare. Per esempio, il vostro computer può estrarre dei numeri casuali per voi. Digitate il programma sottostante:

```
NEW
10 PRINT RND (0)
20 GOTO 10
RUN
```

RND è il comando per generare numeri casuali. Il loop infinito nel programma sottostante genera numeri casuali all'infinito. Ricordate di fermare il loop col tasto **[Break]**. Per effettuare cambi nel programma, potete listare il programma e utilizzare i tasti del controllo del cursore per inserire caratteri, piuttosto di ridigitare intere linee. Fate esperimenti con i diversi programmi qui sotto descritti:

10 PRINT RND (1)
RUN
10 PRINT RND (123)
RUN
10 PRINT RND (50)
RUN
10 PRINT RND (50000)
RUN

Tutte le vostre modifiche della linea 10 generano numeri casuali tra 0 e 1. Il punto decimale è sempre prima del primo numero in un numero casuale. Un numero casuale molto piccolo che ha un numero sul suo lato sinistro del punto decimale è sempre tra 0 e 1. Ma ci sono alcuni numeri che il computer deve scrivere in notazione scientifica.

Il numero nelle parentesi viene chiamato "variabile muta". Non importa che numero viene usato come variabile muta, ma è importante che le parentesi appaiano e che ci sia incluso qualcosa (qualsiasi numero o lettera). Per digitare facilmente, 0 è solitamente posizionato nella posizione di una variabile muta. Cambiate la linea 10 di nuovo come mostrato sotto:

10 PRINT (RND(0)*10)
RUN
10 PRINT (RND(0)*100)
RUN
10 PRINT (RND(0)*1000)
RUN

Ogni programma genera una differente scala di numeri casuali. PRINT (RND(0)*10) genera numeri maggiori di10 perchè l'istruzione dice al computer di moltiplicare il numero casuale per 10. Moltiplicando per 10 muove il punto decimale di un posto. In PRINT (RND(0)*100), moltiplicando per 100 muove il punto decimale di due posti, in PRINT (RND(0)*1000), moltiplicando per 1000 muove il punto decimale di tre posti. Se volete, potete moltiplicare per un numero più grande per generare numeri casuali più elevati.

Dato che numeri molto grossi con molte cifre dopo il punto decimale non vengono presi in considerazione, il computer ha un'istruzione che gli dice di stampare solo interi. Gli interi sono numeri senza punti decimali. L'istruzione INT dice al computer di eliminare ogni punto decimale. Riprogrammate le tre versioni della linea 10 della pagina precedente e confrontate i risultati:

10 PRINT INT(RND(0)*10)
RUN
10 PRINT INT(RND(0)*100)
RUN
10 PRINT INT(RND(0)*1000)
RUN

Il programma genera numeri nella stessa scala di prima, ma i numeri sono molto leggibili senza le cifre dopo il decimale.

Per generare numeri in un'altra scala specifica, esaminate l'esempio seguente:

10 PRINT INT(RND(0)*3)
RUN
10 PRINT INT(RND(0)*12)
RUN
10 PRINT INT(RND(0)*25)
RUN

Il programma genera numeri casuali che sono sempre di una cifra inferiori per il numero per i quali sono stati moltiplicati. La prima linea 10 genera i numeri 0, 1, e 2. Per generare i numeri casuali 0, 1, 2, e 3 il programma dovrebbe essere scritto in questo modo:

10 PRINT INT(RND(0)*4)
RUN

Per generare solo i numeri 1, 2, e 3, il programma dovrebbe essere cosi' composto:

10 PRINT INT(RND(0)*3)+1
RUN

Per generare tre numeri cominciando da 20, scrivete il seguente programma:

10 PRINT INT(RND(0)*3)+20
RUN

Un Gioco con Numeri Casuali

I programmi con numeri casuali sono molto flessibili. Potete utilizzarli in programmi di giochi col vostro computer. Digitate il seguente programma. Ricordate che per ottenere la freccia trasversale nella linea 5 dovete premere **[Esc]**, tenere premuto **[Shift]** o **[Control]**, e premere **[Clear]**.

```
NEW
1 REM *** NUMBER.GAM ***
5 PRINT " ↖ "
10 SECRETNUM=INT(RND(0)*3)+1
20 PRINT: PRINT "I AM THINKING OF A NUMBER, EITHER
1,2, OR 3. TRY TO GUESS IT."
30 INPUT GUESS
40 IF GUESS=SECRETNUM THEN PRINT "YOU WON."
50 IF GUESS< >SECRETNUM THEN PRINT "YOU LOST."
60 GOTO 10
```

La linea 10 assegna il numero casuale alla variabile numerica chiamata SECRETNUM. La linea 30 permette all'utente di digitare in una domanda e assegnare questo numero alla variabile numerica chiamata GUEST. (Ricordate che le variabili numeriche non hanno bisogno di essere dimensionate al contrario delle variabili stringa.) La linea 40 confronta la risposta col numero segreto. Se sono entrambi uguali il computer stampa:

YOU WON.

La linea 50 inoltre confronta la risposta del numero segreto. Se non è uguale (i simboli significano diverso), il computer stampa:

YOU LOST.

La linea 60 effettua un loop in modo tale da potere giocare di nuovo. (**Il capitolo 6** spiega le istruzioni IF-THEN piu' dettagliatamente).

Programmi Matematici

Le funzioni matematiche del computer possono essere utilizzate sia per lavorare, che per giocare. Se foste uno chef che deve preparare del cibo per un pranzo, avreste bisogno del computer per avere le dosi dei vostri ingredienti. Per esempio, supponete di dover sapere quanti pound di scaloppine dovete acquistare per fare un pranzo a 62 invitati. Sapete per certo che mezzo chilo di scaloppine bastano per 5 persone. Il programma sottostante vi dirà quante scaloppine (espresse in pound - unità di misura inglese) acquistare:

```
NEW
1 REM *** COQUILLE ***
10 PRINT "⇱"
20 GUESTS=62
30 POUNDSTOBUY=1.5/5 * GUESTS
40 PRINT: PRINT "BUY ";POUNDSTOBUY;" POUNDS OF
SCALLOPS."
50 END
```

Il programma produce una risposta (18,6 pounds di scaloppine), ma un calcolatore avrebbe effettuato lo stesso calcolo e la stessa risposta senza questo lavoro. Per farsì che il programma sia più utile aggiungete una variante nel numero di invitati, inserendo un'istruzione INPUT. Digitate la linea aggiuntiva sottostante:

```
15 PRINT: PRINT "HOW MANY GUESTS DO YOU
EXPECT?"
20 INPUT GUESTS
```

Lanciate il programma parecchie volte, inserendo un numero differente di invitati ogni volta. L'ammontare delle scaloppine di cui avete bisogno cambia ogni volta. Per 200 invitati sono richieste 60 pounds di scaloppine; per 430 invitati, 130,8 pounds. La funzione INPUT rende il programma più pratico.

CAPITOLO 6
PRENDERE DECISIONI E RISOLVERE PROBLEMI CON IL BASIC

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

I comandi IF-THEN e FOR-NEXT vi abilitano a scrivere programmi che simulano il modo di approccio alle decisioni o ai problemi umano. Questi comandi, particolarmente usati nei giochi e nelle soluzioni logiche, permettono al programmatore di effettuare scelte per il computer.

Comandi IF-THEN

Per fare della pratica con il comando IF-THEN, digitate il seguente programma:

```
NEW
1 REM *** BRNPROBE.QZ ***
5 PRINT " ⇱ "
10 DIM RAIN$ (3)
20 PRINT: PRINT "YES OR NO, IF IT WERE RAINING
OUTSIDE, WOULD YOU GO OUT WITH AN UMBRELLA"
30 INPUT RAIN$
40 IF RAIN$="YES" THEN PRINT "YOU HAVE A
FORMIDABLE IQ."
50 IF RAIN$="NO" THEN PRINT "YOU ARE A BORN RISK
TAKER."
```

Il Brainprobe Quiz valuta la vostra risposta. Nella linea 40, se la risposta memorizzata nella variabile stringa RAIN$ è yes, il computer stampa il messaggio IQ. Se la risposta è NO, il computer legge la prossima linea, la linea 50, e valuta la variabile stringa RAIN$ ancora, quindi il computer stampa il messaggio. Ad ogni modo, se la vostra risposta non è ne si ne no, il programma finisce. Il programma non ha istruzioni per rispondere a una domanda indefinita. Inserite la linea sottostante:

```
60 GOTO 20
```

Calcoli con IF-THEN

Un'altro modo per incoraggiarvi ad una risposta giusta è fornirvi degli aiuti. Il seguente programma utilizza delle variabili numeriche per indirizzarvi alla risposta:

```
NEW
1 REM *** NUMBER.QZ ***
5 PRINT "↖"
10 SECRETNUM=INT(RND(0)*10)+1
20 PRINT: PRINT "GUESS A SEGRET NUMBER
BETWEEN 1 AND 10."
30 PRINT
40 PRINT "YOUR GUESS";
50 INPUT GUESS
60 PRINT
70 IF GUESS=SECRETNUM THEN PRINT "YOU GOT
IT!":END
80 IF GUESS<SECRETNUM THEN PRINT "TOO LOW.
TRY AGAIN.": GOTO 40
90 IF GUESS>SECRETNUM THEN PRINT "TOO HIGH.
TRY AGAIN.": GOTO 40
```

La linea 80 e 90 calcola la risposta come più grande o come più piccola del numero segreto. Il comando PRINT stampa una frase che vi mette nella condizione di sapere se la risposta deve essere maggiore o minore. I comandi GOTO nella linea 80 e 90 creano un loop infinito se continuate a inserire risposte non corrette.

Fine del Programma

Il Number Quiz è programmato per fermarsi quando scoprite il numero segreto. Quando inserite la risposta correta, la linea 70 da al computer l'istruzione di finire. END ferma il programma e il promt di READY appare sullo schermo.

Errori Trasparenti

Se incidentalmente inserite una lettera al posto di un numero nel vostro tentativo, il computer visualizza un messaggio d'errore, e il programma finisce. Effettuate un errore intenzionale digitando una lettera o premendo solo **[Return]**. Per non considerare l'errore effettuato potete considerare il comando TRAP. Aggiungete la linea sottostante e lanciate il programma di nuovo:

```
45 TRAP 100
100 PRINT: PRINT "PLEASE ENTER A NUMBER ONLY."
110 GOTO 30
```

Nella linea 45, il comando TRAP dice al computer di non fermare il programma quando un errore viene inserito e fa tornare il computer alla linea 100. La linea 100 dice al computer di stampare le direttive per correggere l'errore. La linea 110 fa tornare il computer al punto di partenza. Il comando TRAP va posizionato sempre prima dell'istruzione INPUT, e contiene sempre il numero della linea che risolverà il problema.

Realizzazione di un Quiz con IF-THEN

Un programma può facilmente prevedere delle istruzioni quando la risposta giusta è un numero, inclusa una data. Il seguente programma utilizza l'istruzione IF-THEN e il comando TRAP per salvaguardare i vostri tentativi:

```
NEW
1 REM *** LOVELACE. QZ ***
5 PRINT "↖"
10 PRINT: PRINT "ADA LOVELACE, DAUGHTER OF THE
POET LORD BYRON, WAS MATHEMATICALLY
BRILLANT."
20 PRINT
30 PRINT "IN WHAT YEAR DID SHE WRITE HER
AMAZINGLY ACCURATE DESCRIPTION OF THE FUTURE
USES OF THE COMPUTER";
40 TRAP 200
50 INPUT GUESS
60 IF GUESS=1842 THEN GOTO 100
70 IF GUESS<1842 THEN GOTO 110
80 IF GUESS>1842 THEN GOTO 120
100 PRINT: PRINT "CONGRATULATION! YOU GUESSED
THE YEAR CORRECTLY.":END
110 PRINT: PRINT "THAT WAS TOO EARLY. TRY
AGAIN.":GOTO 20
120 PRINT: PRINT "THAT WAS TOO LATE. TRY AGAIN-
.":GOTO 20
200 PRINT: PRINT "PLEASE ENTER A NUMBER ONLY."
210 GOTO 20
```

Nel Lovelace Quiz, la posizione dei messaggi di PRINT associati con l'istruzione IF-THEN è differente dalla loro posizione nel Brainprobe Quiz e il Number Quiz. Questa differenza dimostra che spesso c'è più di una via per arrivare allo stesso risultato nella programmazione.

Errori del Computer

L'istruzione TRAP permette di avere un controllo più profiquo su Lovelace Quiz, ma non è ancora perfetta. Dato che il computer elabora la data come un numero, accetterà 1842.78 come risposta esatta, ma la risposta esatta è 1842,78. Molti programmi hanno "errori", o problemi. Quando sarete in grado di localizzare gli errori ed eliminarli, avrete imparato a programmare. Ogni principiante incontra degli errori e ne effettua altrettanti. Per divenire un programmatore migliore, studiate questo manuale e forse avrete più esperienza nell'individuare i vostri errori. Imparerete come identificare errori in modo tale da scrivere programmi in modo perfetto.

Il Loop FOR-NEXT: Il Loop Contatore

Ormai avete preso famigliarità con il loop infinito GOTO. Un'altro tipo di loop è il FOR-NEXT. Il loop FOR-NEXT è un loop contatore, che non è indefinito. Digitate NEW e inserite il seguente programma:

```
NEW
10 FOR X=1 TO 4
20 PRINT "POTATO"
30 NEXT X
RUN
```

POTATO appare sullo schermo quattro volte. Cambiate la linea 10 in modo tale da leggerla in questo modo:

```
10 FOR X=1 TO 7
```

Questa volta quando lanciate il programma, lo schermo mostra POTATO sette volte. Il computer effettua un loop sette volte sulla linea 10, 20, e 30. FOR dice al computer quante volte effettuare il loop, e NEXT dice al computer di tornare all'inizio e partire nuovamente. NEXT è simile a GOTO. X è una variabile. Potete utilizzare qualsiasi cosa per rappresentare una variabile. Provate questo nome per la variabile:

```
10 FOR NUM=1 TO 7
30 NEXT NUM
```

Quando lanciate il programma, non ci sono differenze con il programma precedente. Cambiate il nome della variabile di nuovo:

```
10 FOR JKL=1 TO 7
30 NEXT JKL
```

JKL è un nome senza senso per una variabile numerica nel loop FOR-NEXT. Lanciate il programma per vedere che può lavorare ugualmente come prima. Ora inserite questa linea:

15 PRINT JKL,
RUN

L'istruzione PRINT nella linea 15 mostra il valore della variabile. (Inserite la virgola per un chiaro discorso di leggibilità). Ogni volta che il computer ripete il loop FOR-NEXT, la variabile assume il valore del successivo numero nella serie specificata alla linea 10. La prima volta la variabile è 1; la seconda volta è 2; e così via. L'ultimo numero nell'istruzione FOR controlla il numero di volte che il computer ha effettuaro il loop nel programma. Cambiate il numero nella linea 10 come mostrato sotto:

10 FOR JKL=1 TO 50
RUN
10 FOR JKL=1 TO 200
RUN
10 FOR JKL=1 TO 500
RUN

Il Punto di Inizio

Listate il programma. Il primo numero nella linea FOR è il punto d'inizio per il conteggio, e l'ultimo numero è il punto di fine. Anche numeri negativi possono essere inseriti nel punto di partenza per il conteggio. Provate queste variazioni per la linea 10:

10 FOR JKL=1 TO 5
RUN
10 FOR JKL=0 TO 5
RUN
10 FOR JKL=3 TO 5
RUN
10 FOR JKL=10 TO 5
RUN

STEP: Contare Incrementando

Listate il programma, cancellate l'istruzione PRINT nella linea 10 e la virgola nella linea 15, e lanciate il programma. Il computer conta e stampa i numeri molto velocemente. Utilizzate il comando STEP per fare contare il computer incrementando. Provate il programma sottostante:

10 FOR JKL=0 TO 500 STEP 5
RUN
10 FOR JKL=0 TO 500 STEP 2
RUN
10 FOR JKL=0 TO 500 STEP 100
RUN
10 FOR JKL=0 TO 500 STEP 7
RUN

Il computer sarà obbligato a contare ad ogni sequenza che voi specificate.

Contare all'Indietro

Il computer può contare all'indietro se utilizzate il comando STEP-1 e la sequenza appropriata di numeri (dal più grande al più piccolo) per cominciare il conteggio. Per esempio :

10 FOR JKL=500 TO 0 STEP -1
RUN
10 FOR JKL=10 TO 0 STEP -1
RUN
10 FOR JKL=-1 TO -19 STEP -1
RUN

Il computer può contare all'indietro anche incrementando:

10 FOR JKL=500 TO 0 STEP -20
RUN
10 FOR JKL=500 TO 0 STEP -3
RUN
10 FOR JKL=0 TO -500 STEP -50
RUN

Potete anche istruire il computer in modo tale da iniziare e finire ad ogni numero che desiderate:

10 FOR JKL=500 TO 300 STEP -10
RUN
10 FOR JKL=25 TO 0 STEP -1
RUN

Ora sapete come istruire il computer per contare avanti o indietro, per contare incrementando o diminuendo, e come fermasi e ricominciare il conteggio a numeri specifici.

Il Loop "Sandwich" FOR-NEXT

Listate il vostro programma. FOR è sulla linea iniziale, e NEXT è nella linea finale. Tutto ciò che il computer deve fare è un sandwich tra queste istruzioni. Digitate la linea sottostante:

```
10 FOR JKL=1 TO 5
20 PRINT "AVOCADO"
```

Il computer estrarrà l'istruzione o il numero di istruzioni tra i comandi FOR e NEXT che specificano il numero di volte. Fate in modo che il computer stampi queste altre parole:

```
16 PRINT "CHEESE"
17 PRINT "MAYONNAISE"
18 PRINT "MUSTARD"
19 PRINT "TOMATO"
21 PRINT "BACON BITS"
22 PRINT "LETTUCE"
23 PRINT: PRINT
RUN
```

Il computer stampa e conta troppo valocemente per fare in modo che la lettura sullo schermo sia chiara. Non preoccupatevi, stampa l'istruzione PRINT esattamente come voluto nell'istruzione FOR-NEXT. Altre istruzioni, come imputazioni matematiche e istruzioni INPUT, possono anche essere utilizzate nel loop sandwich FOR- NEXT.

Loop Ritardati

Cancellate tutte le istruzioni PRINT in modo tale che non ci sia assolutamente niente nel loop FOR-NEXT al di fuori delle istruzioni FOR e NEXT. Lanciate il programma e guardate cosa accade:

```
15
16
17
18
19
20
21
22
23
LIST
RUN
```

Non succede nulla. Cambiate il numero nella linea 10 e guardate attentamente ancora una volta:

```
10 FOR JKL=1 TO 500
RUN
```

Il promt Ready ha bisogno di qualche secondo per apparire. Cambiate ancora la linea 10:

```
10 FOR JKL=1 TO 5000
```

Questa volta il promt Ready utilizza un tempo considerevolmente maggiore. Il computer sta contando ma non sta stampando i suoi calcoli. Il processo e' simile a un vostro conteggio mentale. Il tempo necessario al promt per apparire sullo schermo è il tempo utilizzato dal computer per contare fino a 5000.

I loop FOR-NEXT sono dispositivi eccellenti per bloccare temporaneamente il computer. In effetti, il loop FOR-NEXT viene utilizzato così frequentemente per le sue funzioni che viene alcune volte chiamato " loop di ritardo" e il nome della variabile comune è DELAY. Riscrivete il loop FOR-NEXT, utilizzando DELAY come il nome di una variabile e numeri diversi nell'istruzione FOR:

```
NEW
10 FOR DELAY=1 TO 300
20 NEXT DELAY
LIST
RUN
```

Alcune volte il loop di ritardo è inserito nella struttura a sandwich nella stessa linea di programma:

```
NEW
10 FOR DELAY=1 TO 300:NEXT DELAY
LIST
RUN
```

Programmi Esempio

I programmi sottostanti utilizzano il loop FOR-NEXT in modi molto differenziati. Il primo programma utilizza il loop FOR-NEXT come un semplice loop di ritardo per lasciare la parola HI sullo schermo in tempo utile per essere letta prima che la linea 30 cancelli lo schermo:

```
NEW
1 REM *** DLAYLOOP ***
5 PRINT " ↖ "
10 PRINT "HI"
20 FOR DELAY=1 TO 800:NEXT DELAY
30 PRINT " ↖ "
40 PRINT "BYE"
50 FOR DELAY=1 TO 800:NEXT DELAY
```

Il successivo programma utilizza una variabile numerica nel loop FOR-NEXT. Utilizza anche un comando TRAP che dice al computer di tornare alla linea precedente, non dandovi specifici messaggi sull'errore:

```
NEW
1 REM *** HOWHIGH? ***
10 DIM A$ (1), HH$ (1)
20 PRINT "↖"
30 PRINT: PRINT "HOW HIGH DO YOU WANT TO
COUNT";
40 TRAP 30
50 INPUT HH
55 HH$=STR$(HH): IF HH$="0" THEN GOTO 30
60 FOR COUNT=1 TO HH
70 PRINT COUNT
80 NEXT COUNT
90 PRINT: PRINT "PLEASE ANSWER (Y/N). WOULD YOU
LIKE TO COUNT AGAIN";
100 TRAP 90
110 INPUT A$
120 IF A$="Y" THEN GOTO 30
130 IF A$="N" THEN PRINT: PRINT "BYE": END
140 GOTO 90
```

L'ultimo programma parafrasa una vecchia canzone rock'n'roll e utilizza dei FOR-NEXT particolari. Questo tipo di loop FOR-NEXT è un loop di ritardo più piccolo all'interno del più grande loop FOR-NEXT. Il programma innotre utilizza OR per creare condizioni multiple nell'istruzione IF-THEN:

```
NEW
1 REM *** CLOCKRCK ***
5 PRINT "↖"
10 FOR X=1 TO 9
20 PRINT X;
30 PRINT "O'CLOCK"
40 FOR DELAY=1 TO 500:NEXT DELAY
50 IF X=3 OR X=6 OR X=9 THEN PRINT "ROCK!":FOR
PAUSE=1 TO 500:NEXT PAUSE
60 NEXT X
70 PRINT: PRINT "WE'RE GOING TO ROCK"
80 PRINT "AROUND THE CLOCK"
90 PRINT "TONIGHT!"
```

CAPITOLO 7
PRODUZIONE DI SUONI E GRAFICI CON IL BASIC

Creare suoni e grafici con alcuni computer è molto complicato, ma non con l'ATARI XE. Il comando SOUND nell'ATARI BASIC, combinato con alcune semplici tecniche di programmazione, è tutto quello di cui avete bisogno. I suoni e i grafici aggiungono nuove dimensioni ai vostri programmi BASIC-tantissime cose nei giochi arcade, trucchi vari, colonne sonore, e alcune visualizzazioni grafiche con alcuni colori.

Disattivazione del Suono

Il vostro ATARI XE può suonare quattro voci alla volta. I quattro souni registrati, o voci, sono numerati 0, 1, 2, e 3. Per selezionare la prima voce digitate SOUND 0; per la seconda, SOUND 1; per la terza, SOUND 2; per la terza, SOUND 3; per la quarta, SOUND 3.

Il comando SOUND nell'ATARI BASIC controlla quattro elementi:

- Voce (0-3)
- Pitch (0-255)
- Distorsione (0-14)
- Volume (0-15)

Il pitch, o frequenza, di un suono è determinato da un numero da 0 a 255, dandovi un totale di 256 frequenze sulle quali scegliere. Il valore di pitch è il secondo numero nel comando SOUND. SOUND 1,50 specifica la seconda voce con un pitch di 50. Assicuratevi che il volume sia attivato sul vostro TV o monitor, quindi digitate:

SOUND 1,50,0,8

Premete **[Return]**. Una grande esplosione, non è vero? Per disattivare il souno, abbassate il volume sul vostro televisore oppure digitate uno dei comandi sottostanti e premete **[Return]**:

END
SOUND 1,0,0,0

La purezza, o la distorsione, è determinata da un numero tra 0 e 14. Nel comando SOUND, la purezza del suono è determinata dal secondo numero. Provate questo:

SOUND 1,50,10,8

Il numero 10 produce un tono puro senza distorsione. Per inserire una piccola distorsione, cambiate il 10 in 06:

SOUND 1,50,06,8

Il computer suona come se fosse in grado di esaurire il suono. Digitate END prima di cominciare.

L'ultimo numero nel comando SOUND controlla il volume. Il numero deve essere tra 0 e 15. Il numero 8 è un buon numero per diversi utilizzi. Fate attenzione a non utilizzare numeri superiori al 12 per non correre il rischio di danneggiare l'altoparlante del vostro TV.

Provate alcune di queste quattro armonie digitando i seguenti comandi:

```
SOUND 0,50,10,8
SOUND 1,100,10,8
SOUND 2,150,10,8
SOUND 3,200,10,8
```

Digitate END per fermare il suono.

Disabilitazione del Suono con Variabili

Le variabili nei comandi SOUND aggiungono versatilità ai vostri programmi. Utilizzando le variabili, potete programmare il vostro computer per cambiare voci, pitch, distorsione e volume dei suoni con sustaine. Inserite e
lanciate il seguente programma:

```
NEW
REM * SET VARIABLES FOR SOUND VALUES
20 VOICE=0:PITCH=100:TONE=8:VOL=8
30 SOUND VOICE,PITCH,TONE,VOL
40 GOTO 20
RUN
```

Per fermare il suono, premete **[Break]** e digitate END. Per sostenere un suono, avete bisogno di ripetere il comando SOUND nel programma. Due metodi comuni sono il loop FOR-NEXT o il loop GOTO come quello nell'esempio successivo. Il seguente programma utilizza un pitch nel loop FOR-NEXT per produrre l'intera gamma di pitch eseguita dal computer:

```
NEW
10 REM * SOUND EFFECTS WITH FOR NEXT LOOP
20 VOICE=0:PITCH=0:TONE=10:VOL=8
30 FOR PITCH=0 TO 255
40 SOUND VOICE, PITCH, TONE, VOL
50 NEXT PITCH
RUN
```

Variando il volume in un programma produce una varietà di suoni. Cambiate VOL=8 in VOL=0 e premete **[Return]**. Quindi aggiungete la seguente linea:

35 VOL=INT(RND(0)*16)

Questa linea seleziona in modo casuale un valore variabile tra 0 e 15 per il volume. lanciate il programma per vedere in che modo gli effetti del suono relativi al volume cambiano in modo casuale.

Fare Musica

Il comando SOUND può produrre toni musicali. La seguente scala include note musicali e i loro pitch:

Nota	Pitch	Nota	Pitch
C alto	29	B	64
B	31	A	72
A	35	G	81
G	40	F	91
F	45	E	96
E	47	D	108
D	53	C medio	121
C	60		

Digitate e lanciate il seguente programma:

```
NEW
10 REM ** SIMPLE SONG
15 DIM PITCH$ (1)
20 VOICE=0:PITCH=0:TONE=10:VOL=8
30 REM ** C=121:D=108:E=96:F=91
40 TRAP 300
50 PRINT "↖"
60 PRINT " NOTES FOR SIMPLE SONG"
65 FOR NOTE=1 TO 8
70 READ PITCH
80 SOUND VOICE,PITCH,TONE,VOL
90 GOSUB 200
100 PRINT: PRINT PITCH$
110 FOR PAUSE=1 TO 500:NEXT PAUSE
120 SOUND 0,0,0,0
130 NEXT NOTE
140 GOTO 300
150 REM ** DATA FOR NOTES
160 DATA 121,121,108,96,96,91,108,121
```

```
200 REM **PRINT NOTES
210 IF PITCH=121 THEN PITCH$="C"
220 IF PITCH=108 THEN PITCH$="D"
230 IF PITCH=96 THEN PITCH$="E"
240 IF PITCH=91 THEN PITCH$="F"
250 RETURN
300 PRINT: PRINT "END OF SIMPLE SONG":END
RUN
```

I comandi GOSUB-RETURN e READ-DATA permettono al computer di produrre differenti note inserendo una serie di valori per la variabile pitch. GOSUB dice al computer di andare alla "subroutine" che comincia alla linea 200 e continuare dalla linea 250; il comando RETURN rimanda il computer alla linea immediatamente seguente alla linea GOSUB. Il comando READ dice al computer di leggere le voci nalla linea DATA e inserirle nella variabile. Il computer continua nel suo loop attraverso il programma, fino a che i valori delle linee DATA sono esauriti.

Il programma inoltre utilizza il loop FOR-NEXT per determinare la lunghezza delle note. Utilizzando differenti loop FOR-NEXT cercate di modificare il programma per produrre note intere, mezze note, e altri tipi di note.

Grafici Colorati

Il vostro ATARI XE ha 16 modalità grafiche che comprendono 256 colori. Per cominciare vi diciamo questo capitolo descrive le differenti modalità grafiche e alcune dei comandi grafici più utilizzati.

I 16 colori base e i loro valori corrispondenti sono mostrati sotto. (I colori variano a seconda dei controlli del colore e luminosità sul vostro televisore.)

0 Grigio
1 Oro
2 Arancione
3 Rosso-Arancione
4 Rosa
5 Viola
6 Rosso-Arancione
7 Blu
8 Blu
9 Blu elettrico
10 Turchese
11 Verde-Blu
12 Verde
13 Giallo-Verde
14 Arancio-Verde
15 Arancio chiaro

I rimanenti 112 colori sono ottenuti addizionando un colore per la luminosità e il contrasto.

I registri del colore sono un altro elemento importate della grafica ATARI. I registi del colore possono essere utilizzati nei disegni. Ogni registro può contenere uno dei 128 colori. Dato che sono cinque registri, un massimo di cinque colori diversi può essere visualizzato. I cinque registri del colore sono numerati 0, 1, 2, 3, e 4.

SETCOLOR è un comando grafico essenziale. Il formato è SETCOLOR 2,10,8: il primo numero è il registro del colore; il secondo è il numero del colore; e il terzo è la luminosità.

Modo Grafico 0

I registri del colore funzionano differentemente nei modi grafici utilizzati. La loro funzione nel modo grafico 0 (il modo testo) sono mostrati nella seguente tabella:

Colori di Default	**Registro**	**Funzione**
	0	Non utilizzato
Blu Elettrico	1	Luminosità del testo
Blu Scuro	2	Contorno
	3	Non utilizzato
Nero	4	Bordo

I colori di default sono colori che il computer utilizza automaticamente fin quando voi non lo istruite con colori diversi. Utilizzate SETCOLOR per cambiare colore, digitate il seguente comando:

SETCOLOR 2,3,4

Quando premete **[Return]**, lo schermo diviene arancione. La trasformazione del colore viene effettuata dato che il colore 2 rappresenta il colore dello schermo, il 3 è color arancio e il 4 indica la luminosità. Cambiate il 4 in 6. L'arancione cambia in un arancio chiaro. Cambiate il 6 col 7. Non succede niente dato che i colori tra 0 e 14 definiscono la luminosità. Se digitate un'altro numero, il computer utilizza il colore del precedente numero inserito. Cambiate il 7 in 8 e guardate il colore che diventa più chiaro. Il seguente programma visualizza i 128 colori e la relativa luminosità:

```
NEW
10 REM ** 128 ATARI COLORS
20 REM ** 16 COLORS
30 FOR COLOR =0 TO 15
40 REM ** 8 LUMINANCES
50 FOR LUMINANCE=0 TO 14 STEP 2
60 SETCOLOR 2,COLOR,LUMINANCE
65 PRINT "COLOR=";COLOR;"LUMINANCE=";LUMINANCE
70 REM ** PAUSE TO SEE COLOR
80 FOR PAUSE=1 TO 600:NEXT PAUSE
90 NEXT LUMINANCE
100 NEXT COLOR
RUN
```

Quando la luminosità raggiunge il numero 10, il testo scompare dato che la luminosità di default del testo è 10. (la luminosità di default è la luminosità che il computer automaticamente utilizza sin quando non lo istruite di cambiarla.) Tutte le volte che la luminosità di sfondo è la stessa di quella del testo, il testo sembra scomparire. Fate attenzione al colore di sfondo e alla luminosità del testo tutte le volte che lavorate con i colori e la luminosità nella modalità grafica 0. Digitate GR.0 (che è un'abbreviazione del modo grafico 0) per ripristinare i colori normali dello schermo.

Cambiate SETCOLOR 2 con SETCOLOR 4 nella linea 60 e lanciate il programma ancora. Dato che i registri 4 controllano il bordo, questa volta cambia il colore del bordo piuttosto che quello dell'area di sfondo. Digitate GR.0 per ripristinare i normali colori dello schermo.

Modi Grafici 1 e 2

I modi grafici 1 e 2 vi danno la possibilità di utilizzare un'enorme quantità di colori relativi al testo. Il modo grafico 2 è uguale al modo grafico 1 ad eccezione che ogni carattere è doppio. Il modo grafico 1 ha 24 linee orizzontali e il modo grafico 2 ne ha 12. Per inserire il modo grafico 1, digitare i seguenti comandi:

```
NEW
10 GRAPHICS 1
20 PRINT #6;"GRAPHICS MODE ONE"
```

Lanciate il programma. Il modo grafico 1 è un testo arancione nella parte alta dello schermo. Nella parte bassa è una linea blu contenente la parola READY. La linea blu è la finestra del testo e visualizza il testo in modo grafico 0. Digitate GR.0 per tornare al nuovo testo.

Per stampare un'enorme quantità di testo sullo schermo nel modo grafico 1 e 2, utilizzate PRINT#6: Seguite l'esempio e quindi il testo che volete stampare. Questa istruzione è una variazione del comando PRINT che avete imparato precedentemente.

Ora listate il programma. Cambiate MODE con mode e lanciate il programma. MODE diventa verde. Digitate LIST 20. Utilizzate il tasto di video inverso ◩, cambiate modalità nella linea 20 in MODE e lanciate il progamma. MODE ora cambia in blu. Listate il programma ancora e cambiate MODE con mode e lanciate il programma. Ora MODE è rosso.

Inserite e lanciate il programma seguente:

```
NEW
10 REM ** COLORFULL TEXT
20 GRAPHICS 1
30 PRINT #6;"ORANGE"
40 PRINT #6;"green"
50 PRINT #6;"DARK BLUE "
60 PRINT #6;"red "
70 PRINT "COLORFULL TEXT"
RUN
```

Come potete vedere, il modo grafico 1 è capace di visualizzare cinque colori allo stesso tempo, quattro differenti colori di testo e un colore di fondo. I colori possono essere anche cambiati utilizzando SETCOLOR in accordo alla seguente tabella:

Registro	Colore di Default	Tipo di Cartattere	Colore #	LUM
0	Arancio	Maiuscolo	2	8
1	Verde Chiaro	Minuscolo	12	0
2	Blu Scuro	Maiuscolo inverso	9	4
3	Rosso	Minuscolo inverso	4	6
4	Nero	Sfondo	0	0

Digitate SETCOLOR 4,15,5. Il registro 4 (lo sfondo) cambia in arancio. Ma ora il testo in blu scuro è difficile da leggere. Utilizzate SETCOLOR per cambiarlo. In relazione alla tabella il registro 2 controlla il testo in blu scuro. SETCOLOR 2,8,6 vi permette di fare un trucchetto che fa diventare il blu scuro in un blu un po' più chiaro. Aggiungete le seguenti linee al programma testo:

```
100 FOR COLOR=0 TO 15
110 SETCOLOR 2,COLOR,8
120 FOR DELAY=1 TO 400:NEXT DELAY
130 NEXT COLOR
```

Lanciate il programma. La finestra di testo nella parte bassa dello schermo cambia colore rispetto al blu scuro dato che il registro 2 controlla la finestra di testo allo stesso modo della sua visualizzazione.

Eliminare la Finestra Testo

Alcune volte non avete bisongo di visualizzare la finestra testo nei vostri programmi. Per eliminare la finesta testo, aggiungete 16 al numero della modalità grafica. Cambiate la linea 20 in GRAPHIC 17 e cancellate la linea 70. Il comando PRINT stampa sempre il modo grafico 0. Se siete in modalità 1 o 2, se non avete una finestra aperta, e se utilizzate il comando PRINT e PRINT#6, il computer farà confusione e stamperà tutto in modalità 0. Aggiungete questa linea:

```
70 PRINT "WINDOW TEST"
```

Lanciate il programma e guardate cosa accade. Se utilizzate il comando PRINT e PRINT#6, dovete utilizzate una finestra testo per avere la modalità 1 in modo tale da mostrare quello che è stampato nella parte alta dello schermo.

Cancellate le linee 100, 110, 120 e 130. Lanciate il programma. WINDOW TEST e quindi READY appare in cima allo schermo. Listate il programma. La linea 20 specifica la modalità17 (modalità1 senza la finestra testo), ma dov'è? Sostituite la linea 70 con questa linea:

70 GOTO 70

Quando lanciate il programma, lo schermo in modalità1 ritorna. Quando utilizzate la modalità1 o 2 senza la finestra testo, dovete utilizzare un loop GOTO per mantenere la visualizzazione sullo schermo o vi verrà presentata in modo troppo veloce per essere vista. Premendo **[Break]** ritornate alla modalitá 0.

Per vedere un esempio della modalitá 2, listate il programma e cambiate la linea 20 in:

20 GRAPHICS 18

Graphics 18 stà al posto della modalità 2 più 16 (nessuna finestra testo). Lanciate il programma. Ora avete un testo più colorato.

Per far tornare lo schermo ai suoi colori originali, premete **[Reset]** e digitate SETCOLOR 2,9,4. Non avete perso il programma quando avete premuto **[Reset]** nell'ATARI BASIC. Tuttavia, questa funzione non è applicata in altri linguaggi o programmi.

Modo Grafico 3

Lo schermo in modo grafico 3 è la griglia costituita da 40 colonne e 24 linee (20 se utilizzate la finestra aperta). Inserite e lanciate il seguente programma:

NEW
10 GRAPHICS 3
20 COLOR 1
30 PLOT 0,0
RUN

Nella parte alta a sinistra troverete un quadratino arancione. Il quadrato, o pixel, è un'unità dello schermo grafico. Il comando COLOR determina il colore di un pixel. Il numero dopo il comando COLOR determina quale registro di colore utilizzare per il colore del pixel. Il comando COLOR non mette un colore in un registro; questa operazione viene effettuata da SETCOLOR. Il comando COLOR semplicemente seleziona quale registro utilizzare per disegnare il pixel, e il pixel diventa del colore del registro. Per fare questo in modo chiaro, cambiate la linea 20 in:

20 COLOR 2

Lanciate il programma. Il pixel arancio ora è divenuto verde chiaro. Pensate ogni pixel come un carattere testo. Nella modalità1 e 2, utilizzate caratteri maiuscoli e minuscoli nel modo inverso per selezionare i colori del testo. Nella modalità 3 e ulteriori, utilizzate il comando COLOR per selezionare il colore per il pixel.

PLOT: Disegnare un Punto sullo Schermo

PLOT è simile a PRINT#6; Il comando si differenzia dal fatto che stampa punti al posto di lettere e numeri. COLOR è uguale al metodo di selezione del colore in maiuscolo-minuscolo-video inverso: seleziona il registro. I colori di DEFAULT sonoarancione, verde chiaro, blu scuro e nero. Per cambiare i colori in qualsiasi registro, utilizzare il comando SETCOLOR.

I registri dei colori sono come quattro pennelli. SETCOLOR seleziona il colore che deve essere mandato in ogniuno di questi quattro pennelli, e COLOR seleziona il pennello da utilizzare per disegnare mentre PLOT determina dove il pennello dovrà posizionarsi sullo schermo.

DRAWTO: Collegare i Punti

Aggiungete la linea:

40 DRAWTO 39,0

Lanciate il programma. Una linea verde chiaro apparirà sulla parte alta dello schermo. Dopo aver posizionato il pixel, utilizzate il comando DRAWTO per disegnare un secondo punto e disegnate una linea di collegamento tra di essi. La linea 40 dice al computer di disegnare un pixel alla colonna 39, linea 0, e quindi collegarle. Ora digitate:

DRAWTO 39,19

Il comando disegna un pixel nell'angolo in basso a destra dello schermo grafico, proprio sopra alla linea di testo, e quindi disegna una linea di collegamento tra le coordinate 39.0 e 39.19. Ora digitate:

DRAWTO 0,19

Per completare il rettangolo, digitate:

DRAWTO 0,0

Ora digitate GR.0 e listate il programma. Aggiungete questa linea:

50 DRAWTO 39,19
60 DRAWTO 0,19
70 DRAWTO 0,0

SETCOLOR e COLOR

Quando lanciate il programma, il computer disegna un rettangolo verde di nuovo. Per rendere più luminoso lo schermo, digitate:

35 COLOR 1
45 COLOR 2
55 COLOR 1
65 COLOR 3

Lanciate il programma per guardare un rettangolo di tanti colori.

Per cambiare il colore nel registro, utilizzate SETCOLOR. Potete dedurre che COLOR 1 seleziona il colore del registro 1 e che COLOR 2 seleziona il colore del registro 2. Sfortunatamente questa conclusione non è vera. Il modo 3 ha quattro registri e quattro colori, ma i registri sono numerati 0,1,2 e 4, e i colori sono numerati 0,1,2, e 3. Per capire questo, guardate questa tabella:

Color 0 = Registro 4 Nero
Color 1 = Registro 0 Arancio
Color 2 = Registro 1 Verde chiaro
Color 3 = Registro 2 Blu scuro

Digitate GR.0, listate il programma, e cambiate COLOR 2 nella linea 20 con COLOR 1. COLOR 1 seleziona il registro 0, e il colore di default per il registro 0. Per cambiare il colore nel registro 0, utilizzate il comando SETCOLOR. Aggiungete la seguente linea:

15 SETCOLOR 0,4,6

Quando lanciate il programma, la linea color arancio cambia in un colore simile al rosa. Avete cambiato il colore della linea utilizzando SETCOLOR per cambiare il colore del pennello (il colore nel registro), non utilizzate COLOR per selezionare un differente pennello (registro). La luminosità del colore del registro 0 ha azione sul testo nella finestra di testo.

Ora aggiungete:

42 SETCOLOR 1,2,8

Il verde chiaro sul lato destro del quadrato diventa oro. Aggiungete un'altra linea:

62 SETCOLOR 2,11,4

Lanciate il programma. Ora non è solo il lato sinistro della scatola che cambia in verde, ma anche la finestra di testo diventa verde. In ogni caso, il registro 2 controlla anche il colore della finestra di testo.

Ora dovreste essere in grado di utilizzare il tasto SETCOLOR e COLOR per memorizzare una grossa quantità di colori e utilizzarli nei vostri programmi.

Modi Grafici 5 e 7

Le differenze tra i modi 3, 5, e 7 possono essere illustrate molto semplicemente. Cambiate la linea 10 con:

10 GRAPHICS 5

Lanciate il programma. Il rettangolo è molto più piccolo perchè i pixel sono più piccoli. Con la finestra di testo, la griglia del modo 3 ha 39 colonne e 20 righe. Il modo 5 ha 80 colonne e 40 righe.

Ora cambiate la linea 10 in:

10 GRAPHICS 7

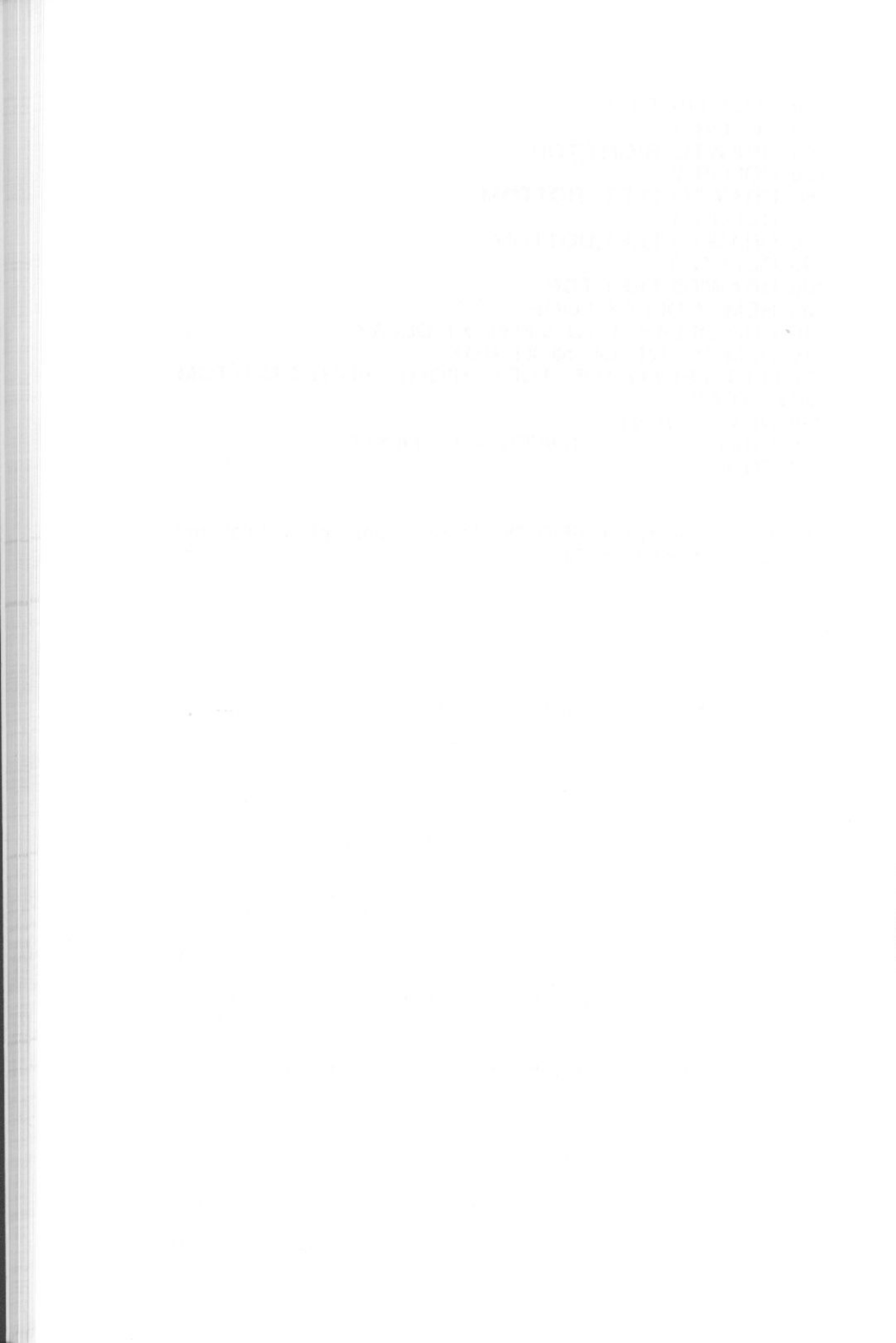
Quando lanciate il programma, qualsiasi rettangolo appare più piccolo. la griglia nella modalità 7 ha 160 colonne per 80 linee.

I pixel più piccoli, determinano una risoluzione più elevata. Delle tre modalità, la modalità 3 è la più bassa e la modalità 7 la più alta. Cercate di disegnare un rettangolo attorno al bordo dello schermo nelle modalità 5 e 7.

Il seguente programma illustra tutto quello che avete trovato in questa sezione. Digitate e lanciate questo:

```
NEW
5 REM ** BILL'S BOX (PLOT AND DRAW)
10 PRINT "WHICH MODE (3,5, OR 7)";
20 LEFT=0:TOP=0
30 INPUT MODE
40 IF MODE=3 THEN RIGHT=39:BOTTOM=19
50 IF MODE=5 THEN RIGHT=79:BOTTOM=39
60 IF MODE=7 THEN RIGHT=159:BOTTOM=79
70 GRAPHICS MODE
80 PRINT "          GRAPHICS MODE ";MODE
90 FOR COUNT=1 TO 1000
100 COLOR 2
110 TRAP 240
115 REM ** DRAW BOX
```

```
120 PLOT LEFT,TOP
130 COLOR 1
140 DRAWTO RIGHT,TOP
150 COLOR 2
160 DRAWTO LEFT, BOTTOM
170 COLOR 1
180 DRAWTO LEFT,BOTTOM
190 COLOR 3
200 DRAWTO LEFT,TOP
205 REM ** DELAY LOOP
210 FOR DELAY=1 TO 500:NEXT DELAY
215 REM ** SIZE OF NEXT BOX
220 LEFT=LEFT+2:TOP=TOP+2:RIGHT=RIGHT-2:BOTTOM
=BOTTOM-2
230 NEXT COUNT
240 PRINT "           THAT'S ALL FOLKS!"
250 END
```

Provate ad utilizzare SETCOLOR per cambiare i colori nel programma Scatola di Bill.

APPENDICE A
SEMPLICI PROGRAMMI IN BASIC

Il vostro ATARI XE può fare miracoli con un piccolo aiuto della vostra immaginazione e le giuste tecniche di programmazione. Questi programmi esempio vi mostreranno la versatilità del vostro ATARI XE e vi motiveranno nello scrivere alcuni programmi da soli.

Ora inserite il programma esattamente come scritto, premendo **[Return]** alla fine di ogni linea. Quando avete finito, digitate la parola RUN, premete **[Return]**, e guardate il vostro ATARI XE vivere.

Nota: quando gli spazi nella linea di un programma sono difficili da identificare, una nota alla fine del programma specificherà l'esatto numero di spazi necessari.

IL CHOO-CHOO ATARI

Gli effetti sonori sono una specialità dell'ATARI. Se chiudete i vostri occhi quando lanciate l'ATARI CHOO-CHOO, potreste pensare di essere su un espresso per Marrakesh.

```
10 POKE 764,255:POKE 580,1
20 GRAPHICS 17:POKE 712,148:POSITION 1,10:PRINT #6;
"THE ATARI CHOO-CHOO"
30 FOR X=15 TO 0 STEP-1-P:SOUND 1,0,0,X
40 R=INT(RND(0)*300)+1
50 IF R=30 THEN SOUND 3,36,10,10:SOUND
2,48,10,10:GOSUB 90:SOUND 3,0,0,0:SOUND 2,0,0,0
60 NEXT X:P=P+0.03
70 IF P>=5 THEN P=5
80 GOTO 30
90 POKE 77,0:POSITION 8,12:PRINT #6; "toot":FOR A=1
TO 400: NEXT A:POSITION 8,12: PRINT #6;"    ": RETURN
```

Nota: la linea 90 richiede quattro spazi bianchi tra gli apici.

IL BIG BANG

Chiudete la porta prima di lanciare il seguente programma in modo tale da non disturbare i vicini.

```
10 POKE 764,255:POKE 580,1
20 GRAPHICS 17
30 FOR X=10 TO 100:SOUND 0,X,10,10:SOUND
1,X-2,10,8:SOUND 2,X+2,10,12:NEXT X
40 SOUND 1,0,0,0:SOUND 2,0,0,0
50 POSITION 4,11:PRINT #6; "BAROOOOMMM!"
60 FOR DECAY=15 TO 0 STEP-0.5:FOR B=1 TO 20:
SOUND 0,100,B,DECAY:POKE 712,B:NEXT B:NEXT
DECAY
70 GRAPHICS 1+32:POKE 712,148
80 POKE 752,1:PRINT: PRINT " Press start to set off
another explosion."
90 IF PEEK(53279)<>6 THEN GOTO 90
100 GOTO 20
```

CERCATE QUESTE PAROLE

Questo programma di ricerca inserisce parole negli spazi appropriati, in ordine alfabetico. Mettete le parole nelle istruzioni data nelle linee 10 e 20 per ordinare le parole da voi scelte. Ricordare di separare ogni parola con una virgola.

```
10 DATA ATARI,DISK DRIVE,MONITOR,COMPUTER,
TOUCH TABLET,PRINTER,KEYBOARD
20 DATA SOFTWARE,PROGRAM RECORDER,WORD
PROCESSING,ACCOUNTING,DATA BASE,FUN
30 DIM Z$(1000),A(50),A$(20),S(10)
40 S(1)=1:FOR L=1 TO 9:S(L+1)=S(L)*3+1:NEXT L
50 TRAP 80:GRAPHICS 0:? "HERE IS THE LIST:"
60                                        READ
A$:B=LEN(Z$):C=LEN(A$):Z$(B+1,B+1)=CHR$(C):? A$
70 Z$(B+2,B+1+C)=A$:Q=Q+1:A(Q)=B+1:GOTO 60
80 ?:? "READY TO SORT...",:P=0
90 P=P+1:IF S(P+2)<Q THEN 90
100 FOR I=P TO 1 STEP-1:S=S(I):FOR J=S+1 TO Q:L=
J-S:A=A(J): B=A(L)
110 IF Z$(A+1,A+ASC(Z$(A,A)))>Z$ (B+1,B+ASC(Z$(B,B)))
THEN 130
120 A(L+S)=B:L=L-S:IF L>0 THEN B=A(L):GOTO 110
130 A(L+S)=A:NEXT J:NEXT I:? : ? "SORTED."
140 FOR L=1 TO Q:A=A(L): ? Z$(A+1,A+ASC(Z$(A,A))):
NEXT L
```

PLAYERS AND MISSILES

Questo programma utilizza una tecnica chiamata Grafica a Missili per Giocatori per creare un mostro rosa che si muove lungo lo schermo di fronte a una linea verticale di colore blu. Se volete che il mostro si muova al di fuori della barra blu, semplicemente cambiate la linea 150 in 150 POKE 623,4.

```
10 POKE 764,255:POKE580,1
20 GRAPHICS 3+16
30 FOR X=16 TO 24:FOT Y=0 TO 23: COLOR 3 PLOT
X,Y:NEXT Y:NEXT X
40 MEMTOP=PEEK(741)+256*PEEK(742)-1
50 PMBASE=INT((MEMTOP-1024)/1024)*1024
60 ADJTOP=PMBASE+384
70 POKE742,INT(ADJTOP256):POKE741,ADJTOP-256*
PEEK(742)
80 POKE 54279,PMBASE/256
90 POKE 53277,2
100 POKE 559,34+8
110 PO=PMBASE+512
120 FOR A=PO TO PO+128:POKE A,0:NEXT A
130 FOR A=PO+60 TO PO+67:READ B:POKE A,B:NEXT A
140 POKE 53256,3
150 POKE 623,1
160 POKE 704,108
170 POKE 53248,PEEK(20):GOTO 170
180 DATA 60,126,129,153,255,36,66,129
```

TOPSY-TURVY

Quando lanciate Topsy-Turvy, il vostro schermo sarà riempito da strane scritte. Per controllarle, semplicemente premete **[Start]**. Per rifare le cose di nuovo, premete **[Select]**.

```
10 POKE 764,255:POKE 580,1
20 GRAPHICS 18:POKE 712,128: POKE 755,5
30 POSITION 5,3:PRINT #6; "WELCOME TO"
40 POSITION 2,5:PRINT #6; "THE TOPSY-TURVY":
POSITION 6,7: PRINT #6; "WORLD OF":POSITION 6,9
50 PRINT #6; "COMPUTERS"
60 IF PEEK(53279)=5 THEN POKE 755,5:POKE 712,128
70 IF PEEK(53279)=6 THEN POKE 755,1:POKE 712,99
80 GOTO 60
```

DIGITARE UNA MELODIA

Questo programma assegna valori alle note musicali in relazione ai tasti della parte alta della tastiera. Premete solo un tasto alla volta.

TASTO	VALORE MUSICALE
[Insert]	B
[Clear]	B^b (o A#)
0	A
9	A^b (o G#)
8	G
7	F^b (o G#)
6	F
5	E
4	E^b (o D#)
3	D
2	D^b (o C#)
1	C

```
10 DIM CHORD(37),TUNE(12)
20 GRAPHICS 0:? :? "      TYPE-A-TUNE PROGRAM"
25 ?:? "PRESS KEYS 1-9,0,<,> TO PRODUCE NOTES.";
27 ?:? "RELEASE ONE KEY BEFORE PRESSING THE
NEXT."
28 ?:? "OTHERWISE, THERE MAY BE DELAY."
30 FOR X=1 TO 37:READ A:CHORD(X)=A:NEXT X
40 FOR X=1 TO 12:READ A:TUNE(X)=A:NEXT X
50 OPEN #1,4,0, "K:"
55 OLDCHR =-1
60 A=PEEK(764):IF A=255 THEN 60
63 IF A=OLDCHR THEN 100
65 OLDCHR=A
70 FOR X=1 TO 12:IF TUNE(X)=A THEN SOUND
0,CHORD(X),10,8:GOTO 100
80 NEXT X
100 I=INT(PEEK(53775)/4):IF (I/2)=INT(I/2) THEN 60
110 POKE 764,255:SOUND 0,0,0,0:OLDCHR=-1 :GOTO 60
200 DATA243,230,217,204,193,182,173,162,153,144,136,
128,121,114,108,102,96,91,85,81,76,72,68,64,60
210 DATA 57,53,50,47,45,42,40,37,35,33,31,29
220 DATA 31,30,26,24,29,27,51,53,48,50,54,55
```

Per suonare " Mary Had a Little Lamb ", premete i seguenti tasti:

[5][3][1][3][5][5][5] [3][3][3] [5][8][8]
[5][3][1][3][5][5][5] [5][3][3][5][3][1]

Nota: Assicuratevi di inserire tre spazi tra THE e NEXT nella linea 27.

ALTA MATEMATICA

Il vostro computer ATARI è un calcolatore molto versatile. Quando inserite due numeri nel programma sottostante, il computer vi dirà qual'è il Massimo Comun Denominatore. Per esempio, se inserite 690911 e 11214017, scoprirete che il Massimo Comun Denominatore è 53147.

```
10 ? CHR$(125):? "Enter two numbers. Press Return after
each entry."
20 INPUT N1,N2
30 GOSUB 90
40 ? "Their GCD is ";:?AN
50 POKE 752,1:POSITION 10,10: ? "Press Start to
continue."
60 IF PEEK(53279)< >6 THEN GOTO 60
70 POKE 752,0:? CHR$(125) :GOTO 10
80 REM ****SUBROUTINE****
90 AN=0:POKE195,0:TRAP130:M=(N1>=N2)*N1+(N2>N1)*
N2:N=(M=N1)*N2+(M=N2)*N1
100 IF INT(N1)< >N1 OR INT(N2)< >N2 THEN RETURN
110 P=M-INT (M/N)*N:M=N:N=P
120 IF P< >0 THEN GOTO 110
130 AN=M*(PEEK(195)=0):RETURN
```

COMPUTER BLUES

Questo programma genera note musicali casuali per "scrivere" alcune melodie molto interessanti per il basso programmato.

```
1 GRAPHICS 0:? :? "    COMPUTER BLUES":?
2 PTR=1
3 THNOT=1
5 CHORD=1
6 PRINT "BASS TEMPO (1=FAST)";
7 INPUT TEMPO
8 GRAPHICS 2+16:GOSUB 2000
10 DIM BASE(3,4)
20 DIM LOW(3)
25 DIM LINE(16)
26 DIM JAM(3,7)
30 FOR X=1 TO 3
40 FOR Y=1 TO 4
50 READ A:BASE(X,Y)=A
60 NEXT Y
70 NEXT X
80 FOR X=1 TO 3:READ A:LOW(X)=A
90 NEXT X
95 FOR X=1 TO 16:READ A:LINE(X)=A:NEXT X
96 FOR X=1 TO 3
97 FOR Y=1 TO 7
98 READ A:JAN(X,Y)=A:NEXT Y:NEXT X
100 GOSUB 500
110 T=T+1
115 GOSUB 200
120 GOTO 100
200 REM PROCESS HIGH STUFF
205 IF RND(0)<0.25 THEN RETURN
210 IF RND(0)<0.5 THEN 250
220 NT=NT+1
230 IF NT>7 THEN NT=7
240 GOTO 260
250 NT=NT-1
255 IF NT<1 THEN NT=1
260 SOUND 2,JAM(CHORD,NT),10,NT*2
280 RETURN
500 REM PROCESS BASE STUFF
510 IF BASS=1 THEN 700
520 BDUR=BDUR+1
530 IF BDUR< >TEMPO THEN 535
531 BASS=1:BDUR=0
535 SOUND 0,LOW(CHORD),10,4
540 SOUND 1,BASE(CHORD,THNOT),10,4
550 RETURN
700 SOUND 0,0,0,0
```

```
710 SOUND 1,0,0,0
720 BDUR=BDUR+1
730 IF BDUR< >1 THEN 800
740 BDUR=0:BASS=0
750 THNOT=THNOT+1
760 IF THNOT< >5 THEN 800
765 THNOT=1
770 PTR=PTR+1
780 IF PTR=17 THEN PTR=1
790 CHORD=LINE(PTR)
800 RETURN
1000 DATA 162,144,136,144,121,108,102,108,108,96,91,96
1010 DATA 243,182,162
1020 DATA 1,1,1,1,2,2,2,2,1,1,1,1,3,2,1,1
1030 DATA 60,50,47,42,40,33,29
1040 DATA 60,50,45,42,40,33,29
1050 DATA 81,68,64,57,53,45,40
2000 PRINT #6:PRINT #6:PRINT #6
2005 PRINT #6; "      Computer"
2006 PRINT #6
2010 PRINT #6; "          Blues"
RETURN
```

LA BANDIERA DEGLI STATI UNITI D'AMERICA

Questo programma utilizza la selezione dei colori per disegnare le linee. Utilizza il modo grafico 7 più 16 in modo tale che la visualizzazione appaia a tutto schermo. Si noti la corrispondenza delle istruzioni COLOR con le istruzioni SETCOLOR. Per divertirvi e sperimentare, aggiungete un istruzione SOUND e utilizzate una combinazione di READ/DATA per aggiungere "The Star Spangled Banner" dopo la linea 470.

```
10 REM DRAW THE UNITED STATES FLAG
20 REM HIGH RESOLUTION 4-COLOR GRAPHICS, NO
TEXT WINDOW
30 GRAPHICS 7+16
40 REM SETCOLOR 0 CORRESPONDS TO COLOR 1
50 SETCOLOR 0,4,4:RED=1
60 REM SETCOLOR 1 CORRESPONDS TO COLOR 2
70 SETCOLOR 1,0,14:WHITE=2
80 REM SETCOLOR 2 CORRESPONDS TO COLOR 3
90 BLUE=3:REM DEFAULTS TO BLUE
100 REM DRAW 13 RED & WHITE STRIPES
110 C=RED
120 FOR I=0 TO 12
130 COLOR C
140 REM EACH STRIPE HAS SEVERAL HORIZONTAL
LINES
150 FOR J=0 TO 6
160 PLOT 0,I*7+J
170 DRAWTO 159,I*7+J
180 NEXT J
190 REM SWITCH COLOR
200 C=C+1:IF C>WHITE THEN C=RED
210 NEXT I
300 REM DRAW BLUE RECTANGLE
310 COLOR BLUE
320 FOR I=0 TO 48
330 PLOT 0,I
340 DRAWTO 79,I
350 NEXT I
360 REM DRAW 9 ROWS OF WHITE STARS
370 COLOR WHITE
380 K=0:REM START WITH ROW OF 6 STARS
390 FOR I=0 TO 8
395 Y=4+I*5
400 FOR J=0 TO4:REM 5 STARS IN A ROW
410 X=K+5+J*14:GOSUB 1000
420 NEXT J
430 IF K<>0 THEN K=0:GOTO 470
440 REM ADD 6TH STAR EVERY OTHER LINE
450 X=5+5*14:GOSUB 1000
```

```
460 K=7
470 NEXT I
500 REM IF KEY HIT THEN STOP
510 IF PEEK(764)=255 THEN 510
515 REM OPEN TEXT WINDOW WITHOUT CLEARING
SCREEN
520 GRAPHICS 7+32
525 REM CHANGE COLOR BACK
530 SETCOLOR 0,4,4:SETCOLOR 1,0,14
550 STOP
1000 REM DRAW 1 STAR CENTERED AT X,Y
1010 PLOT X-1,Y:DRAWTO X+1,Y
1020 PLOT X,Y-1:PLOT X,Y+1
1030 RETURN
```

IGPAY ATINLAY

Questo breve programma converte parole o frasi in Latino. Fate però attenzione a non inserire una parola di una sola lettera coma A o I.

```
10 DIM A$(256):S=2
20 ? "Type in a word or sentenge. Please don't exceed three
lines of text."
30 INPUT A$
40 FOR X=1 TO LEN(A$)
50 IF A$(X,X)=CHR$(32) THEN PRINT A$(S,X-
1);A$(S-1,S-1); "AY";" ";:S=X+2
60 IF X=LEN(A$) THEN PRINT A$(S,X); A$(S-1,S-1); "AY"
70 NEXT X
80 ? :? :? "THAT'S ALL FOLKS!"
```

GRAPHEEK

Digitate questo programma e guardate il suo effetto grafico.

```
10 DIM A$(35)
20 GRAPHICS 1
25 TRAP 90
30 A$="THIS IS A GRAPHICS DEMOSTRAZION."
40 FOR I =1 TO 33:? #6;A$(I,I);
50 S=PEEK(53770)
60 SOUND 0,S,10,14
70 FOR DELAY=0 TO 100:NEXT DELAY
80 NEXT I
90 SOUND 0,0,0,0:END
```

Nota: Assicuratevi di inserire due spazi tra GRAPHICS e DEMONSTRATION nella linea 30.

ESREVER

Il titolo di questo programma è semplicemente la parola REVERSE stampata al contrario. Per stampare parole pronunciate al contrario, digitate questo breve programma. Dopo averlo lanciato un punto di domanda apparirà sul vostro schermo. Inserite una parola o una breve frase e lasciate che il vostro ATARI XE faccia il lavoro.

```
10 DIM A$(180)
20 PRINT "Enter a word or short sentence and press Return."
30 INPUT A$
40 FOR X=LEN(A$) TO 1 STEP-1
50 PRINT A$(X,X);
60 NEXT X
70 PRINT :PRINT :GOTO 20
```

PROTEZIONE DEL VOSTRO PROGRAMMA

Una domanda da porsi che è molto importante è la seguente. Come proteggere i programmi da occhi indiscreti e dita veloci? Una serie di linee di programma vi potranno aiutare nel tenere lontani occhi indiscreti dai vostri programmi.

In primo luogo provate con questo programma:

```
10 FOR X=1 TO 50:POKE 710,X: NEXT X:GOTO 10
```

Per proteggere il programma, aggiungete un'altra linea di programma per disabilitare il tasto **[Break]**. Questa linea protegge il programma da qualsiasi persona che voglia fermarlo o listarlo durante la sua esecuzione. Inoltre, se costruite un programma che richiede l'inserimento della tastiera, disabilitando il tasto **[Break]** lo proteggete da quella azione che viene chiamata in gergo "finger slip", che puó fermare il programma attraverso il tasto **[Break]** in modo tale da sproteggere il programma.

Cancellate il GOTO 10 dal programma colori e aggiungete questa linea:

```
20 POKE 16,64:POKE 53774,64:GOTO 10
```

Ora lanciate il vostro nuovo programma e cercate di fermarlo premendo il tasto **[Break]**. Non potete utilizzarlo.

Per avere gli affetti desiderati, l'istruzione POKE deve essere inserita nel vostro programma dopo ogni comando che abilita un modo grafico.

La disabilitazione del tasto **[Break]** ha i suoi limiti. Alcuni esperti programmatori vi potranno mostrare di poter interrompere il vostro programma e listarlo, premendo semplicemente il tasto **[Reset]**. Per evitare questa operazione digitate questa linea di programma:

5 POKE 580,1

Quando l'ospite non voluto preme il tasto **[Reset]**, il programma colori viene eliminato dalla memoria del computer nessun programma, nessun listato! L'istruzione POKE dovrebbe essere inserita sempre all'inizio del vostro programma.

GABBIANI SOPRA L'OCEANO

Questo programma unisce grafica e suoni. I suoni non sono suoni puri, simulano infatti il rumore dell'oceano e il canto dei gabbiani. Per ottenere il simbolo alla linea 20 utilizzate **[Control][G], [Control][F], [Control][R], [Control][R]**.

```
10 DIM BIRD$(4)
20 BIRD$="\ / - - "
30 FLAG=1:ROW=10:COL=10
40 GRAPHICS 1:POKE 756,226:POKE 752,1
50 SETCOLOR 0,0,0:SETCOLOR 1,8,14
60 PRINT #6;"         the ocean"
70 R=INT(RND(0)*11)
80 POSITION 17,17
90 FOR T=0 TO 10
100 SOUND 0,T,8,4
110 FOR A=1 TO 50:NEXT A
120 IF RND(0)>0.8 THEN FOR D=10 TO 5 STEP-1:SOUND
1,0,10,INT(RND(0)*10):NEXT D:SOUND 1,0,0,0
130 GOSUB 200
140 NEXT T
150 FOR T=10 TO 0 STEP-1
160 SOUND 0,T,8,4
170 FOR A=1 TO 50:NEXT A
175 IF RND(0)>0.8 THEN FOR D=10 TO 5 STEP-1:SOUND
1,D,10,8:NEXT D:SOUND 1,0,0,0
180 FOR H=1 TO 10:NEXT H
185 GOSUB 200
190 NEXT T
195 GOTO 70
200 GOSUB 300
210 POSITION COL,ROW
220 PRINT #6;BIRD$(FLAG,FLAG+1)
230 FLAG=FLAG+2:IF FLAG=5 THEN FLAG=1
240 RETURN
300 IF RND(0)>0.5 THEN RETURN
310 POSITION COL,ROW
320 PRINT #6;"  "
330 A=INT(RND(0)*3)-1
340 B=INT(RND(0)*3)-1
350 ROW=ROW+A
360 IF ROW=0 THEN ROW=1
370 IF ROW=20 THEN ROW=19
380 COL=COL+B
390 IF COL=0 THEN COL=1
400 IF COL>18 THEN COL=18
410 RETURN
```

Nota: Due spazi sono richiesti alla linea 320.

KINETIC ART

Mettete i colori in movimento con il programma che crea un arcobaleno di linee in continuo movimento.

```
10 REM KINETIC BY NEIL HARRIS
20 GRAPHICS 10
30 DIM A(3,50)
35 FOR L=0 TO 3:FOR M=0 TO 50:A(L,M)=0:NEXT M:NEXT
L
40 HUE=INT(RND(1)*8+1): POKE704+HUE,INT(RND(1)*8)*
16+INT(RND(1)*4+4)
50 X1=INT(RND(1)*80*:X2=INT(RND(1)*80):Y1=INT(RND(1)
*192)
60  COLOR0:PLOTA(0,WHICH),A(1,WHICH):DRAWTO
A(2,WHICH),A(3,WHICH)
70 BOUNCE=BOUNCE-1:IF BOUNCE>0 THEN 90
80 BOUNCE=INT(RND(1)*10+10):BX1=INT(RND(1)*9-4):
BX2=INT(RND(1)*9-4):BY1=INT(RND(1)*13-6):BY2=
INT(RND(1)*13-6)
90 CHANGE=CHANGE-1:IF CHANGE>0 THEN 110
100 CHANGE=INT(RND(1)*10+5):HUE=INT(RND(1)*8+1):
POKE 704+HUE,INT(RND(1)*256)
110 COLOR HUE:PLOT X1,Y1:DRAWTO X2,Y2
120 A(0,WHICH)=X1:A(1,WHICH)=Y1:A(2,WHICH)=X2:A (3,
WHICH)=Y2
130 WHICH=WHICH+1:IF WHICH>50 THEN WHICH=0
140 X1=X1+BX1:IF X1<0 OR X1>79 THEN BX1= -BX1:
GOTO 140
150 X2=X2+BX2:IF X2<0 OR X2>79 THEN BX2= -BX2:
GOTO 150
160 Y1=Y1+BY1:IF Y1<0 ORY1>191 THEN BY1=BY1:
GOTO 160
170 Y2=Y2+BY2:IF Y2<0 OR Y2>191 THEN BY2= -BY2:
GOTO 170
180 GOTO 60
```

APPENDICE B
PARORE RISERVATE IN BASIC

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Nota: Il punto è ingiuntivo dopo tutte le parole chiave abbreviate.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
ABS		Riporta il valore assoluto di una variabile o di un espres sione.
ADR		Riporta l'indirizzo di memoria di una variabile stringa.
AND		Funziona come un operatore logico. L'espressione è vera solo se entrambe le sottoespressioni seguite da AND sono vere.
ASC		Riporta il valore numerico di una singola stringa di caratteri.
ATN		Riporta l'arcotangente di un numero o di una espressione in radianti o gradi.
BYE	B.	Esce dal BASIC e vi riporta al sistema operativo residenteo al processore della console.
CLOAD	CLOA.	Carica dati dal registratore alla RAM.
CHR$		Riporta i byte di una singola stringa equivalenti a un numero tra 0 e 255 nel codice ATASCII.
CLOG		Esprime il logaritmo in base 10 di una espressione.
CLOSE	CL.	Chiude tutti i file alla conclusione di una operazione I/O. Funziona come un comando I/O.
CLR		E' l'opposto della funzione DIM e unidimensiona tutte le matrici e le stringhe.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
COLOR	C.	Sceglie il registro del colore da essere utilizzato nel lavoro grafico.
COM		Ha le stesse funzioni di DIM.
CONT	CON.	Significa "continua". Il programma ricomincia l'esecuzione alla linea successiva sulla quale è stato fermato dal tasto **[Break]** o per aver incontrato STOP.
COS		Esprime il coseno di una variabile o di una espressione in gradi o radianti.
CSAVE		Estrapola i dati dalla RAM al registratore per il salvataggio su nastro.
DATA	D.	Come parte della combinazione READ-DATA, identifica il succedersi delle voci (che devono essere separate da virgole) come voci individuali stringhe.
DEG	DE.	Dice al computer di effettuare funzioni trigonometriche in gradi al posto di radianti. (La misurazione di DEFAULT in radianti.)
DIM	DI.	Riserva l'ammontare di memoria specificato per matrici, array e variabili stringa.(Tutte le variabili stringa, gli arrays e le matrici devono essere dimensionate con il comando DIM.)
DOS	DO.	Significa "Disk Operating System". Serve per visualizzare il menu. (Fare riferimento al manuale DOS.)
DRAWTO	DR.	Disegna una linea retta tra un punto disegnato e un altro specificato.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
END		Ferma l'esecuzione del programma; chiude i files; disattiva il suono. Può essere usato più di una volta in un programma. (CONT può essere usato per far ripartire il programma.)
ENTER	E.	Memorizza il dato o il programma in una forma non tokenizzata (sorgente). Funziona come un comando I/O.
EXP		Esprime una elevazione ad una data potenza e (2.7182818).
FOR	F.	Utilizzato con NEXT per stabilire il loop FOR-NEXT. Introduce la scala che la variabile del loop opererà durante l'esecuzione del loop stesso.
FRE		Riporta l'ammontare di memoria in byte rimasta.
GET	GE.	Utilizzato frequentemente con operazioni su disco per inserire un singolo byte di dati.
GOSUB	GOS.	Salta all'inizio della subroutine al numero di linea specificato.
GOTO	G.	Salto incondizionato al numero di linea specificato.
GRAPHICS	GR.	Specifica uno dei 16 modi grafici. (GR.0 può essere usato per pulire lo schermo.)
IF		Causa salti condizionati o l'esecuzione di un'altra istruzione sulla stessa linea (solo se la prima espressione è vera).

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
INPUT	I.	Obbliga il computer a domandarvi un inserimento dalla tastiera. L'esecuzione continua solo quando il tasto **[Return]** viene premuto dopo che il dato è stato inserito. Funziona anche come comando I/O.
INT		Specifica il valore intero arrotondato per eccesso di un valore. L'arrotondamento è per difetto anche quando il numero è negativo.
LEN		Specifica la lunghezza della stringa specificata in byte o caratteri. (un byte contiene un carattere.)
LET	LE.	Assegna un valore al nome della variabile specificata. (LET è opzionale nel BASIC ATARI e può essere omesso.)
LIST	L.	Visualizza su video o d'uscita il listato del programma.
LOAD	LO.	Caricamento dal disco al computer.
LOCATE	LOC.	Memorizza in una variabile specificata il valore che con trolla un punto grafico specificato.
LOG		Specifica il logaritmo naturale di un numero.
LPRINT	LP.	Controlla la linea della stam pante per specificare un messaggio.
NEW		Cancella tutto il contenuto della RAM utente.
NOT		Riporta un 1 solo se l'espressione non è vera; riporta uno 0 se è vera.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
NEXT	N.	Utilizzato nel loop FOR-NEXT per terminare o continuare. In base alle particolari variabili o espressioni.Tutti i loop sono eseguiti alla fine una volta.
NOTE	NO.	Utilizzato solo in operazioni con disco. (vedi manuale DOS.)
ON		Utilizzato con GOTO o GOSUB per operazioni di salto. (sono possibili salti multipli a differenti numeri di linea, dipende dal valore della variabile o dell'espressione in ON.)
OPEN	O.	Apre il file specificato per operazioni di INPUT o OUTPUT.
OR		Utilizzato come operatore logico tra due espressioni. Se una delle due è vera, viene specificato un 1, se entrambe sono false il risultato è 0.
PADDLE		Controlla la posizione del paddle.
PEEK		Riporta la forma decimale del contenuto di una locazione di memoria specificata (RAM o ROM).
PLOT	PL.	Disegna un singolo punto alle locazioni specificate X,Y.
POINT	P.	Utilizzato solo in operazioni con disco (vedi manuale DOS).
POKE	POK.	Inserisce il byte specificato nella locazione di memoria stabilita. Può essere usato solo con la RAM.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
POP		Cancella le variabili del loop dallo stack del GOSUB. Utilizzato quando la partenza del loop viene effettuata in modo differente dal normale.
POSITION	POS.	Posiziona il cursore ne punto specificato dello schermo.
PRINT	PR. o ?	Trasferisce dal computer al dispositivo d'uscita specificato. Funziona come un comando I/O.
PTRIG		Riporta lo stato del pulsante del fuoco in un joystick.
PUT	PU.	Manda un singolo byte di dati dal computer alla periferica specificata.
RAD		Dice al computer di fornire informazioni in radianti piutto sto che in gradi, per funzioni trigonometriche. (L'unità di misura di default è in radianti. Vedi DEG.)
READ	REA.	Legge le voci nella lista di DATA e assegna alla variabile specificata.
REM	R.	Significa nota. Non fà assolutamente niente ma permette di inserire dei commenti nel listato del programma da utilizzarsi come riferimento per il programmatore. L'istruzione REM non è eseguita.
RESTORE	RES.	Permette ai dati di essere letti più di una volta.
RETURN	RET.	Fa tornare il computer da una subroutine all'istruzione immediatamente seguente quella nella quale appare GOSUB.
RND		Ritorna un numero casuale tra 0 e 1; mai 1.

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
RUN	RU.	Esegue il programma; posiziona le variabili normali a 0, unidimensiona gli arrays e le stringhe.
SAVE	S.	Forza i dati e i programmi ad essere registrati su disco sotto il file specificato con SAVE. Funziona come un comando I/O.
SETCOLOR	SE.	Memorizza i colori e la luminosità in un particolare registro del colore.
SGN		Ritorna +1 se il valore è positivo, 0, se è zero; -0 se è negativo.
SIN		Ritorna il seno trigonometrico di un dato valore in gradi o radianti.
SOUND	SO.	Controlla i registri del pitch, della distorsione e del volume di un tono o di una nota.
SQR		Riporta la radice quadrata di un valore specificato.
STATUS	ST.	Richiama la routine di stato per un particolare dispositivo.
STEP		Utilizzato con FOR-NEXT. Determina la quantità di salti per ogni valore della variabile del loop.
STICK		Riporta la posizione del joystick.
STRIG		Riporta un 1 se il tasto del joystick non è premuto; uno 0 se premuto.
STOP	STO.	Causa l'arresto del programma ma non chiude i file o disattiva il suono.
STR$		Riporta una stringa carattere al valore numerico dato. (Per esempio, STR$(65) riporta a 65 come una stringa.)

PAROLA RISERVATA	ABBREVIAZIONE	BREVE SOMMARIO DELLE ISTRUZIONI BASIC
THEN		Utilizzato con IF. Se la espressione è vera, il comando THEN viene eseguito. Se l'espressione è falsa il controllo passa allalinea successiva.
TO		Utilizzato con FOR, come in "FOR X=1 TO 10." Separa le espressioni della scala del loop.
TRAP	T.	Prende il controllo del programma nel caso di errore in INPUT e direziona l'esecuzione al numero di linea specificato.
USR		Riporta il risultato di una subroutine in linguaggio macchina.
VAL		Riporta l'equivalente valore numerico di una stringa.
XIO	X.	Utilizzato con operazioni su disco (vedi manuale DOS) e nei lavori grafici. Funziona come una istruzione I/O.

APPENDICE C
SET DI CARATTERI ATASCII

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
0	0	♥	Control ,	á
1	1	┣	Control A	ù
2	2	┃	Control B	Ñ
3	3	┛	Control C	É
4	4	┫	Control D	Ç
5	5	┓	Control E	ô
6	6	╱	Control F	ò
7	7	╲	Control G	ì
8	8	◢	Control H	£
9	9	▗	Control I	ï
10	A	◣	Control J	ü
11	B	▝	Control K	ä
12	C	▘	Control L	Ö
13	D	▔	Control M	ú
14	E	▁	Control N	ó
15	F	▖	Control O	ö
16	10	♣	Control P	Ü
17	11	┏	Control Q	â

Note

ATASCII sostituisce ATARI ASCII. Le lettere e i numeri Hanno lo stesso valore di quelle in ASCII, ma alcuni caratteri speciali sono differenti.

Fanno eccezione come mostrato, i caratteri da 128 a 255 che sono i colori inversi dei caratteri da 1 a 127.

Aggiungete 32 al codice maiuscolo per ottenere il codice minuscolo della stessa lettera.

Per ottenere il codice ATASCII, dite al computer (in modo diretto) di stampare ASC ("___"). Riempite gli spazi bianchi con una lettera o un carattere. Dovete utilizzare gli apici.

La normale visualizzazione dei simboli grafici è mostrata con simboli bianchi su sfondo nero; i simboli inversi sono mostrati come simboli neri su sfondo bianco.

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
18	12	─	Control R	û
19	13	┼	Control S	î
20	14	●	Control T	é
21	15	▄	Control U	è
22	16	▎	Control V	ñ
23	17	┬	Control W	ê
24	18	┴	Control X	å
25	19	▌	Control Y	à
26	1A	└	Control Z	Å
27	1B	Esc	Esc Esc	
28	1C	↑	Esc Control -	
29	1D	↓	Esc Control =	
30	1E	←	Esc Control +	
31	1F	→	Esc Control *	
32	20		Space bar	
33	21	!	Shift 1	
34	22	"	Shift 2	
35	23	#	Shift 3	
36	24	$	Shift 4	
37	25	%	Shift 5	
38	26	&	Shift 6	
39	27	'	Shift 7	
40	28	(	Shift 9	
41	29	)	Shift 0	
42	2A	*	*	
43	2B	+	+	
44	2C	,	,	
45	2D	-	-	
46	2E	.	.	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
47	2F	/	/	
48	30	0	0	
49	31	1	1	
50	32	2	2	
51	33	3	3	
52	34	4	4	
53	35	5	5	
54	36	6	6	
55	37	7	7	
56	38	8	8	
57	39	9	9	
58	3A	:	Shift ;	
59	3B	;	;	
60	3C	<	<	
61	3D	=	=	
62	3E	>	>	
63	3F	?	Shift /	
64	40	@	Shift 8	
65	41	A	A	
66	42	B	B	
67	43	C	C	
68	44	D	D	
69	45	E	E	
70	46	F	F	
71	47	G	G	
72	48	H	H	
73	49	I	I	
74	4A	J	J	
75	4B	K	K	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
76	4C	L	L	
77	4D	M	M	
78	4E	N	N	
79	4F	O	O	
80	50	P	P	
81	51	Q	Q	
82	52	R	R	
83	53	S	S	
84	54	T	T	
85	55	U	U	
86	56	V	V	
87	57	W	W	
88	58	X	X	
89	59	Y	Y	
90	5A	Z	Z	
91	5B	[	Shift ,	
92	5C	\	Shift +	
93	5D	]	Shift .	
94	5E	^	Shift *	
95	5F	_	Shift –	
96	60	♦	Control ·	
97	61	a	a	
98	62	b	b	
99	63	c	c	
100	64	d	d	
101	65	e	e	
102	66	f	f	
103	67	g	g	
104	68	h	h	
105	69	i	i	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
106	6A	j	j	
107	6B	k	k	
108	6C	l	l	
109	6D	m	m	
110	6E	n	n	
111	6F	o	o	
112	70	p	p	
113	71	q	q	
114	72	r	r	
115	73	s	s	
116	74	t	t	
117	75	u	u	
118	76	v	v	
119	77	w	w	
120	78	x	x	
121	79	y	y	
122	7A	z	z	
123	7B	♠	Control .	Ä
124	7C	|	Shift =	
125	7D	↖	Esc Control < or Esc Shift <	
126	7E	◀	Esc Delete Bk Sp	
127	7F	▶	Esc Tab	
128	80	♥	◤ Control .	
129	81	┣	◤ Control A	
130	82	▕	◤ Control B	
131	83	┛	◤ Control C	
132	84	┫	◤ Control D	
133	85	┓	◤ Control E	
134	86	╱	◤ Control F	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
135	87		Control G	
136	88		Control H	
137	89		Control I	
138	8A		Control J	
139	8B		Control K	
140	8C		Control L	
141	8D		Control M	
142	8E		Control N	
143	8F		Control O	
144	90		Control P	
145	91		Control Q	
146	92		Control R	
147	93		Control S	
148	94		Control T	
149	95		Control U	
150	96		Control V	
151	97		Control W	
152	98		Control X	
153	99		Control Y	
154	9A		Control Z	
155	9B	EOL	Return	
156	9C		Esc Shift Delete Bk Sp	
157	9D		Esc Shift >	
158	9E		Esc Control Tab	
159	9F		Esc Shift Tab	
160	A0		Space bar	
161	A1		Shift 1	
162	A2		Shift 2	
163	A3		Shift 3	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
164	A4	$	Shift 4	
165	A5	%	Shift 5	
166	A6	&	Shift 6	
167	A7	'	Shift 7	
168	A8	(	Shift 9	
169	A9	)	Shift 0	
170	AA	*	*	
171	AB	+	+	
172	AC	,	,	
173	AD	-	-	
174	AE	.	.	
175	AF	/	/	
176	B0	0	0	
177	B1	1	1	
178	B2	2	2	
179	B3	3	3	
180	B4	4	4	
181	B5	5	5	
182	B6	6	6	
183	B7	7	7	
184	B8	8	8	
185	B9	9	9	
186	BA	:	Shift ;	
187	BB	;	;	
188	BC	<	<	
189	BD	=	=	
190	BE	>	>	
191	BF	?	Shift /	
192	C0	@	Shift 8	
193	C1	A	A	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
194	C2	B	◤ B	
195	C3	C	◤ C	
196	C4	D	◤ D	
197	C5	E	◤ E	
198	C6	F	◤ F	
199	C7	G	◤ G	
200	C8	H	◤ H	
201	C9	I	◤ I	
202	CA	J	◤ J	
203	CB	K	◤ K	
204	CC	L	◤ L	
205	CD	M	◤ M	
206	CE	N	◤ N	
207	CF	O	◤ O	
208	D0	P	◤ P	
209	D1	Q	◤ Q	
210	D2	R	◤ R	
211	D3	S	◤ S	
212	D4	T	◤ T	
213	D5	U	◤ U	
214	D6	V	◤ V	
215	D7	W	◤ W	
216	D8	X	◤ X	
217	D9	Y	◤ Y	
218	DA	Z	◤ Z	
219	DB	[	◤ Shift ,	
220	DC	\	◤ Shift +	
221	DD	]	◤ Shift .	
222	DE	^	◤ Shift *	
223	DF	_	◤ Shift -	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
224	E0	♦	Control .	
225	E1	a	a	
226	E2	b	b	
227	E3	c	c	
228	E4	d	d	
229	E5	e	e	
230	E6	f	f	
231	E7	g	g	
232	E8	h	h	
233	E9	i	i	
234	EA	j	j	
235	EB	k	k	
236	EC	l	l	
237	ED	m	m	
238	EE	n	n	
239	EF	o	o	
240	F0	p	p	
241	F1	q	q	
242	F2	r	r	
243	F3	s	s	
244	F4	t	t	
245	F5	u	u	
246	F6	v	v	
247	F7	w	w	
248	F8	x	x	
249	F9	y	y	
250	FA	z	z	
251	FB	♠	Control ;	
252	FC	|	Shift =	
253	FD	↖	Esc Control 2	

CODICE DECIMALE	CODICE ESADECIMALE	CARATTERE ATASCII	TASTO	CARATTERE EUROPEO
254	FE		Esc Control Delete Bk Sp	
255	FF		Esc Control >	

APPENDICE D
MESSAGGI D'ERRORE

CODICE	MESSAGGIO
2	INSUFFICIENT MEMORY. (Memoria insufficiente). Non è rimasta sufficiente memoria RAM per memorizzare l'istruzione o il nome della nuova variabile, o per dimensionare una nuova variabile striga con array.
3	VALUE ERROR. (Errore di valore). Un valore che doveva essere un intero positivo è negativo; un valore che non è tra la gamma specificata.
4	TOO MANY VARIABLES. (Troppe variabili). Il numero massimo dei nomi delle variabili accettato è 128.
5	STRING LENGHT ERROR. (Errore di lunghezza della stringa). L'utente ha tentato di memorizzare le variabili stringa che eccedono al dimensionamento della lunghezza della stringa.
6	OUT OF DATA. (Al di fuori dei dati). L'istruzione READ richiede più voci di quelle fornite nelle istruzioni DATA.
7	LINE NUMBER GREATER THAN 32767. (Numero di linea più alto di 32767). Il numero di linea è più grande di 32767.
8	INPUT STATEMENT ERROR. (Errore di inserimento del comando). L'utente ha tentato di inserire un valore non numerico in una variabile numerica.
9	ARRAY OR STRING DIM ERROR. (Errore nel dimensionamento dell'array o della stringa). La dimensione del DIM ha superato il numero 5460 per l'array numerico o 32767 per le stringhe, un array o una stringa è stato ridimensionato; il riferimento è stato fatto ad un array o una stringa unidimensionati.
11	FLOATING POINT OVERFLOW/UNDERFLOW. (La divisione non ha avuto buon fine). L'utente ha cercato di dividere per zero o ha tentato di riferirsi a un numero più grande di $1*10^{98}$ o $1*10^{-98}$.

12 LINE NOT FOUND.(Linea non trovata). Un GOSUB, GOTO o THEN ha fatto riferimento a un numero di linea non esistente.

13 NO MATCHING FOR STATEMENT. (Comando FOR senza effetto). Un NEXT è stato incontrato senza un preventivo FOR, oppure non si è arrivati a una giusta posizione del FOR/NEXT. (Il messaggio d'errore è riportato all'istruzione NEXT, ma non al FOR).

14 LINE LENGTH ERROR. (Errore di lunghezza della linea). L'istruzione è troppo complessa o troppo lunga per il BASIC.

15 GOSUB OR FOR LINE DELETED. (Linea GOSUB o FOR cancellata). Un istruzione RETURN o NEXT è stata incontrata, ma il corrispondente GOSUB o FOR è stato cancellato dall'ultimo comando RUN.

16 RETURN ERROR. (Errore di RETURN). Un RETURN è stato incontrato senza un GOSUB.

17 SYNTAX ERROR.(Errore di sintassi). Il computer ha incontrato una linea con errore di sintassi.

18 INVALID STRING CHARACTER. (Stringa di carattere non valida). La stringa nell'istruzione VAL non è una stringa numerica.

I seguenti errori sono errori INPUT/OUTPUT (I/O) che risultano, durante l'utilizzo del disk drives, della stampante, o di altre periferiche.Ulteriori informazioni vengono date dall' hardware ausiliario.

19 LOAD PROGRAM TOO LONG. (Programma troppo lungo per il caricamento). La memoria rimanente è insufficiente per completare il LOAD.

20 DEVICE NUMBER ERROR. (Errore numerico del dispositivo). Il numero del dispositivo è più grande di 7 o è uguale a 0.

21 LOAD FILE ERROR. (Errore di caricamento del file).L'utente ha cercato di caricare un file non caricabile, cioè un file BASIC non tokenizzato. I file tokenizzati sono creati con il comando SAVE.

128 BREAK ABORT. (Interruzione). L'utente ha premuto il tasto **[BREAK]** durante un operazione di I/O.

129 IOCB* ALREADY OPEN. (IOCB già aperto). Il controllo di blocco di INPUT/OUTPUT è già aperto.

130 NONEXISTENT DEVICE. (Periferica non esistente). L'utente ha cercato di accedere a una periferica non definita (per esempio, un dispositivo non contemplato).

131 IOCB* WRITE-ONLY ERROR. (Errore di sola scrittura in IOCB). Un comando READ è stato mandato a un dispositivo a sola scrittura (stampante).

132 INVALID COMMAND. (Comando non valido). Il comando non è valido per questo dispositivo.

133 DEVICE OR FILE NOT OPEN. (Dispositivo o file non aperto). Nessun comando OPEN è stato specificato per il dispositivo.

134 BAD IOCB* NUMBER. (Numero IOCB non valido). Il numero della periferica non è esatto.

135 IOCB* READ-ONLY ERROR. (Errore di sola lettura IOCB). Un comando WRITE è stato mandato a una periferica a sola lettura.

136 EOF. (Fine del file). Il computer ha raggiunto la fine del file.

137 TRUNCATED RECORD. (Record troncato). Questo errore appare tipicamente quando il record che è stato letto è più grande della massima dimensione del record specificata nella chiamata alla CIO. (La dimensione massima del record nell'ATARI BASIC è 119 byte.)

138 DEVICE TIME-OUT. (Errore del dispositivo). La periferica non risponde.

139 DEVICE NAK. (NAK del dispositivo). I problemi sono localizzati alla porta seriale o nella periferica.

140 SERIAL BUS INPUT FRAMING ERROR. L'informazione è stata persa dalla periferica al computer.

*IOCB si riferisce al blocco di controllo INPUT/OUTPUT. Il numero di dispositivo è lo stesso del numero di IOCB.

141 CURSOR OUT OF RANGE. (Cursore fuori scala). Il cursore è fuori dalla scala per una modalità particolare.

142 SERIAL BUS DATA FRAME OVERRUN. (L'informazione è stata persa dalla periferica al computer).

143	SERIAL BUS DATA FRAME CHECKSUM ERROR. L'informazione è stata persa dalla periferica al computer.
144	DEVICE DONE ERROR. L'utente ha cercato di scrivere sul dischetto protetto dalla scrittura.
145	READ AFTER WRITE COMPARE ERROR L'utente ha cercato di aprire l'Editor di schermo con un numero riferito ad un modo grafico non giusto.
146	FUNCTION NOT IMPLEMENTED. La funzione non è stata implementata nel calcolatore.
147	INSUFFICIENT RAM. Non c'è abbastanza memoria per operare nel modo grafico selezionato.
160	DRIVE NUMBER ERROR. L'utente ha specificato il numero di drive sbagliato.
161	TOO MANY OPEN FILES. Non è disponibile il settore del buffer.
162	DISK FULL. Non ci sono settori liberi disponibili.
163	UNRECOVERABLE SYSTEM I/O ERROR. La versione del DOS sul disco può essere danneg giata.
164	FILE NUMBER MISMATCH. Il file su disco può essere danneggiato.
165	FILE NAME ERROR. La specificazione del file ha nel suo interno caratteri non contemplati.
166	POINT DATA LENGTH ERROR. Il secondo parametro dell'istruzione POINT è troppo grande.
167	FILE LOCKED. L'utente ha cercato di accedere a un file protetto per effettuare operazioni al di fuori della lettura.
168	INVALID COMMAND. Il comando in un codice di operazione specifica non è valido.
169	DIRECTORY FULL. L'utente ha utilizzato tutti gli spazi aperti (64 nomi dei file) nella directory.
170	FILE NOT FOUND. L'utente ha cercato di accedere ad un file non esistente nella directory del dischetto.
171	INVALID POINT. L'utente ha cercato di utilizzare POINT relativo a un byte in un file non aperto per l'aggiornamento.

172 ILLEGAL APPEND. L'utente ha cercato di utilizzare il DOS II per aprire un file DOS I per l'aggiornamento. Il DOS II non può essere aperto dai file DOS I. Utilizzate il DOS II, per copiare i files DOS I nel dischetto DOS II.

173 BAD SECTORS AT FORMAT TIME. Il disk drive ha trovato dei settori non buoni durante la formattazione del dischetto. Utilizzate un altro dischetto perchè un dischetto con settori non buoni non può essere formattato. Se questo errore si ripete più volte su un dischetto, il disk drive può avere bisogno di riparazione.

INDICE ALFABETICO

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

Consulenza per gli utenti

L'Atari Corporation è felice di rispondere hai quesiti riguardanti i vostri apparecchi Atari. Scrivete a:

Atari Corporation
Customer Relations
P.O. Box 61657
Sunnyvale, CA 94088

In Gran Bretagna, scrivete a:

Atari Corp. (UK) Ltd.
Customer Relations
P.O. Box 555
Slough
Berkshire
SL2 5BZ

In Italia, scrivete a:

Atari Italia S.p.A
Via Bellini, 21
20095 - Cusano Milanino (MI)

Vi saremo grati se scrivete l'oggetto della vostra lettera sulla busta.

Gli "Atari user groups" attendono notizie su come ottenere il massimo dai vostri apparecchi Atari. Per ricevere l'elenco degli "Atari user groups" della vostra zona scrivete a:

Atari Italia S.p.A
Via Bellini, 21
20095 - Cusano Milanino (MI)